

Oracle® Communications

Cloud Native Core Console User Guide



Release 22.4.1

F74470-02

January 2023

ORACLE®

Copyright © 2019, 2023, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1	Introduction	
	CNC Console Compatibility Matrix	1-1
	Audience	1-2
	Reference	1-2
2	CNC Console Features	
	Support for Network Functions	2-1
	Support for OCCNE Common Services	2-6
	LDAP Integration	2-6
	SAML 2.0 Integration	2-7
	Logging Support	2-8
	Support for Multi Cluster Deployment	2-10
	cnDBTier Integration	2-12
	TLS Certificate Support	2-12
	Service Mesh Communication	2-13
3	Logging into CNC Console	
	CNC Console Dashboard	3-1
4	Configuring CNC Console IAM	
	Role Based Access Control in CNC Console IAM	4-1
	Types of Roles in CNC Console	4-1
	Accessing Roles in CNC Console Applications	4-2
	Creating or Updating Admin Password in CNC Console IAM	4-3
	Creating or Updating User Password in CNC Console IAM	4-4
	Configuring the CNC Console Redirection URL	4-6
	Users in CNC Console IAM	4-7
	Creating the Users	4-8
	Viewing the Users	4-9
	Assigning Roles to the User	4-10

Integrating SAML SSO with CNC Console IAM	4-11
Integrating CNC Console LDAP Server with CNC Console IAM	4-18
Configuring User Federation with CNC Console IAM	4-20
Grouping the LDAP Mapper and Assigning the Roles	4-25

5 Configuring NF Resources in CNC Console IAM

Generate Access Tokens	5-1
Refresh Access Tokens	5-4
Accessing NF Resources	5-4

6 REST APIs and HTTP Response Codes

Generate access token	6-1
Create a new user	6-2
Get users	6-3
Get single user	6-5
Delete the User	6-6
Set up a New Password for the User	6-7
Get Realm Level Roles	6-9
Get Realm-level Role Mappings	6-10
Add Realm-level Role Mappings to the User	6-11
Remove All User Sessions Associated with the User	6-13
Get the Client details	6-14
Update the Client	6-15

7 CNC Console Metrics

CNC Console IAM Metrics	7-2
CNCC IAM Requests	7-2
CNCC IAM Response	7-2
CNCC IAM Success Responses	7-3
CNCC IAM 5xx Responses	7-3
CNCC IAM 4xx Responses	7-4
CNCC IAM Error Responses	7-4
CNCC IAM Access Token Request	7-5
CNCC IAM Access Token Granted	7-6
CNCC IAM Access Token Not Granted	7-6
CNCC IAM User Login Failure Responses	7-7
CNC Console Core Metrics	7-8
CNCC Core Requests	7-8
CNCC Core Responses	7-9

CNCC Core Success Responses	7-9
CNCC Core 5xx Responses	7-10
CNCC Core 4xx Responses	7-10
CNCC Core Error Responses	7-11
CNCC Core Access Token Request	7-11
CNCC Core Access Token Granted Responses	7-12
CNCC Core Access Token Not Granted Responses	7-13
CNCC Core User Login Failure Responses	7-14
CNCC Core User Authorization Failure Responses	7-15
CNCC Core BSF Requests	7-16
CNCC Core BSF Responses	7-17
CNCC Core DD Requests	7-17
CNCC Core DD Responses	7-18
CNCC Core NRF Responses	7-18
CNCC Core NRF Requests	7-19
CNCC Core NSSF Requests	7-20
CNCC Core NSSF Responses	7-20
CNCC Core POLICY Requests	7-21
CNCC Core POLICY Responses	7-21
CNCC Core SCP Responses	7-22
CNCC Core SCP Requests	7-22
CNCC Core SEPP Requests	7-23
CNCC Core SEPP Responses	7-23
CNCC Core UDR Requests	7-24
CNCC Core UDR Responses	7-25

8 CNC Console Alerts

CNC Console IAM Alerts	8-1
CncclamTotalIngressTrafficRateAboveMinorThreshold	8-1
CncclamTotalIngressTrafficRateAboveMajorThreshold	8-2
CncclamTotalIngressTrafficRateAboveCriticalThreshold	8-3
CncclamMemoryUsageCrossedMinorThreshold	8-4
CncclamMemoryUsageCrossedMajorThreshold	8-4
CncclamMemoryUsageCrossedCriticalThreshold	8-5
CncclamTransactionErrorRateAbove0.1Percent	8-6
CncclamTransactionErrorRateAbove1Percent	8-6
CncclamTransactionErrorRateAbove10Percent	8-7
CncclamTransactionErrorRateAbove25Percent	8-7
CncclamTransactionErrorRateAbove50Percent	8-8
CncclamIngressGatewayServiceDown	8-8

CnccIamFailedLogin	8-9
AdminUserCreation	8-9
CnccIamAccessTokenFailure	8-10
CNC Console Core Alerts	8-11
CnccCoreTotalIngressTrafficRateAboveMinorThreshold	8-11
CnccCoreTotalIngressTrafficRateAboveMajorThreshold	8-12
CnccCoreTotalIngressTrafficRateAboveCriticalThreshold	8-12
CnccCoreMemoryUsageCrossedMinorThreshold	8-13
CnccCoreMemoryUsageCrossedMajorThreshold	8-14
CnccCoreMemoryUsageCrossedCriticalThreshold	8-14
CnccCoreTransactionErrorRateAbove0.1Percent	8-15
CnccCoreTransactionErrorRateAbove1Percent	8-16
CnccCoreTransactionErrorRateAbove10Percent	8-16
CnccCoreTransactionErrorRateAbove25Percent	8-17
CnccCoreTransactionErrorRateAbove50Percent	8-17
CnccCoreIngressGatewayServiceDown	8-18
CnccCoreFailedLogin	8-18
CnccCoreUnauthorizedAccess	8-19
CnccCoreAccessTokenFailure	8-20
CNCC Alert configuration in Prometheus	8-20
Validating Alerts	8-22
Disabling Alerts	8-23
Configuring SNMP-Notifier	8-23
MIB Files for CNCC	8-24

9 CNC Console KPIs

CNC Console IAM KPIs	9-1
CNCC IAM Requests	9-1
CNCC IAM Responses	9-1
CNCC IAM Success Rate	9-2
CNCC IAM Error Rate	9-3
CNC Console Core KPIs	9-3
CNCC Core Requests	9-4
CNCC Core Responses	9-5
CNCC Core Success Rate	9-8
CNCC Core Error Rate	9-12
CNC Console Common KPIs	9-16
CPU Usage	9-16
Memory Usage	9-17

10 CNC Console Logs

Log Formats	10-1
Types of Logs	10-4
Examples of Logs	10-8
Accessing logs	10-13

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:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
 - 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**.

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

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

Acronyms

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

Table Acronyms

Acronym	Definition
AD	Active Directory
ASM	Aspen Service Mesh
BSF	Binding Support Function
cnDBTier	Cloud Native DB Tier
CNCC	Cloud Native Core Console
CNE	Cloud Native Environment
OCCNE	Oracle Communications Cloud Native Environment
CS	Common Service
CRUD Operations	CREATE, READ, UPDATE, DELETE
DD	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 OCCNE common service(s) and remote Agent CNC Console (s) (A-CNCC). M-CNCC has two components: M-CNCC IAM and M-CNCC Core.
M-CNCC IAM	Manager CNC Console IAM or M-CNCC IAM (also known as mCncc Iam) is an IAM component of M-CNCC. M-CNCC IAM contains M-CNCC IAM Ingress Gateway and M-CNCC IAM back-end microservices.
M-CNCC Core	Manager CNC Console Core or M-CNCC Core (also known as mCncc Core) is a core component of M-CNCC that provides GUI and API access portal for accessing NF and OCCNE common services. M-CNCC Core contains M-CNCC Core Ingress Gateway and M-CNCC Core back-end microservices.
A-CNCC	Agent CNC Console is a CNCC Core instance which manages local NF(s) and local OCCNE common services(s). A-CNCC is managed by M-CNCC. A-CNCC contains A-CNCC Core Ingress Gateway and A-CNCC Core back-end microservices. A-CNCC has no IAM component. A-CNCC is also known as A-CNCC Core or aCncc Core.
M-CNCC Kubernetes cluster	Kubernetes cluster hosting M-CNCC
mTLS	Mutual Transport Layer Security

Table (Cont.) Acronyms

Acronym	Definition
Instance	OCNF or OCCNE common service managed by either M-CNCC Core or A-CNCC Core.
Site	Kubernetes Cluster
CS	OCCNE Common Services like Grafana, Kibana, Jaeger, Prometheus, Alertmanager and so on.
MC	Multi Cluster. In multi cluster, a single CNCC can manage NF instances that access different Kubernetes clusters.
MO	Managed Objects
MOS	My Oracle Support
LDAP	Lightweight Directory Access Protocol
LDAPS	Lightweight Directory Access Protocol (Over SSL)
NRF	Network Repository Function
OCNF	Oracle Communications Network Function
OSDC	Oracle Software Delivery Cloud
OSO	Operations Stack Overlay
PROVGW	Provisioning Gateway
REST API	Representational State Transfer Application Programming Interface
SCP	Service Communication Proxy
SAML	Security Assertion Markup Language
SBA	Service Based Architecture
SEPP	Security Edge Protection Proxy
TLS	Transport Layer Security
UDR	Unified Data Repository
UE	User Equipment
URI	Subscriber Location Function

What's New in This Guide

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

Release 22.4.1- F74470-02, January 2022

No updates are made in this release.

Release 22.4.0- F74470-01, November 2022

- Updated the [Compatibility Matrix](#) section to support the Data Director (DD), and latest version of NFs.
- Updated the [Support for Network Functions](#) feature section to support the Data Director (DD).
- Updated the [Generating and Refreshing Access Tokens and Accessing NF Resources](#) section. The headers that must be passed for accessing NF resources are updated.
- Updated the [Integrating SAML SSO with CNC Console IAM](#) section. Configuring the CNCC endpoints at SAML IDP and configuring HTTP POST binding methods at CNCC IAM are updated.
- Added the following metrics in the the [CNC Console Metrics](#) section to support the Data Director (DD) NF:
 - CNCC Core DD Requests
 - CNCC Core DD Responses
- Updated the following KPIs in the [CNC Console KPIs](#) section to support the Data Director (DD):
 - CNCC Core Requests
 - CNCC Core Responses
 - CNCC Core Success Rate
 - CNCC Core Error Rate

1

Introduction

The Cloud Native Core Console (CNC Console) is a single screen solution to configure and manage Network Functions (NFs).

The CNC Console has the following modules:

- **CNC Console Core (CNCC Core):** CNCC Core acts as Graphical User Interface (GUI) or Application Programming Interface (API) portal for NFs and Oracle Communications Cloud Native Environment (OCCNE) common services. CNCC Core module is the part of CNC Console that integrates with other cloud native core network functions.
- **CNC Console Identity and Access Management (CNCC IAM):** CNCC IAM acts as local identity provider and as a broker for external identity provider. CNCC IAM module includes the required authentication and authorization procedures such as creating and assigning roles to users.

CNC Console Compatibility Matrix

The following table provides the list of network functions that are compatible with CNC Console 22.4.x:

Release 22.4.1

Table 1-1 Compatibility Matrix

Network Functions	Compatible Versions
BSF	22.4.x
NRF	22.4.x
NSSF	22.4.x
Policy	22.4.x
SCP	22.4.x
SEPP	22.4.x
UDR	22.4.x

Table 1-2 Compatibility Matrix

Components	Compatible Versions
DD	22.0.0
OCCNE	22.2.x, 22.3.x, 22.4.x
cnDBTier	22.2.x, 22.3.x, 22.4.x
CDCS	22.3.x, 22.4.x
OSO	1.6.x, 1.10.x, 22.3.x
ASM	1.4.6-am9, 1.6.14-am4, 1.9.8-am1

Release 22.4.0

Table 1-3 Compatibility Matrix

Network Functions	Compatible Versions
BSF	22.4.0
NRF	22.4.0
NSSF	22.4.0
Policy	22.4.0
SCP	22.4.0
SEPP	22.4.0
UDR	22.4.0

Table 1-4 Compatibility Matrix

Components	Compatible Versions
DD	22.0.0
OCCNE	22.2.x, 22.3.x, 22.4.x
cnDBTier	22.2.x, 22.3.x, 22.4.x
CDCS	22.3.x, 22.4.x
OSO	1.6.x, 1.10.x, 22.3.x
ASM	1.4.6-am9, 1.5.7-am3, 1.6.14-am4, 1.9.8-am1

Audience

The CNC Console User Guide is intended for system administrators, developers, professionals or anyone seeking information about CNC Console.

Reference

Refer to the following documents for more information:

- *Oracle Communications Cloud Native Core Service Communication Proxy User Guide*
- *Oracle Communications Cloud Native Core Network Repository Function User Guide*
- *Oracle Communications Cloud Native Core Policy User Guide*
- *Oracle Communications Cloud Native Core Unified Data Repository User Guide*
- *Oracle Communications Cloud Native Core Binding Support Function User Guide*
- *Oracle Communications Cloud Native Core Security Edge Protection Proxy User Guide*
- *Oracle Communications Cloud Native Core Network Repository Function Installation and Upgrade Guide*
- *Oracle Communications Cloud Native Core Service Communication Proxy Installation Guide*

- *Oracle Communications Cloud Native Core Unified Data Repository Installation and Upgrade Guide*
- *Oracle Communications Cloud Native Core Binding Support Function Installation Guide*
- *Oracle Communications Cloud Native Core Policy Installation Guide*
- *Oracle Communications Cloud Native Core Security Edge Protection Proxy Installation Guide*

2

CNC Console Features

This section explains about the CNC Console features.

Support for Network Functions

CNC Console GUI provides a user interface to configure and manage the following network functions:

Binding Support Function (BSF)

BSF provides the following functions:

- Allows PCF users to register, update, and remove the binding information
- Allows network function consumers to retrieve the binding information

CNC Console provides an interface to configure global and service parameters in BSF.

For more information about configuring parameters, see the *Configuring BSF Using CNC Console* section in *Oracle Communications Cloud Native Core Binding Support Function User Guide*.

Data Director (DD)

Oracle Communications Data Director (OCDD) is a Network Data Broker (NDB) in the 5G core network. OCDD receives the network traffic data from various sources such as 5G NFs, Non-5G nodes, third-party producers and sends the filtered and consolidated data securely to the subscribed consumers, which are 3rd party consumer applications or platforms.

Data collection is a complex task in a 5G Service Based Architecture (SBA). The data collected must be correlated to provide meaningful insights to the customers. The OCDD can filter, replicate, aggregate data and route these data feeds to 3rd party consumers who have subscribed to the feeds. OCDD ensures data security, low latency and redundancy while collecting and processing the data feeds. The OCDD enables the communications service providers (CSP) to correlate and transform the acquired data as per their data feed configuration to create comprehensive dashboards and Key Performance Indicators (KPIs) and thereby achieve meaningful insights about all functions in the 5G network. This information can be used for monetizing, providing good quality of service for the user, reducing downtime, ensuring easy network scalability and minimizing losses. The OCDD generated data feed can be beneficial for monitoring and troubleshooting during a network failure. OCDD is a crucial function which aids in creating self-healing networks.

Network Repository Function (NRF)

OCNRF provides the following functions:

- Maintains the profiles of the available NF instances and their supported services in the 5G core network.

- Allows consumer NF instances to discover other provider's NF instances in the 5G core network.
- Allows NF instances to track the status of other NF instances.
- Provides OAuth2 based Access Token service for consumer NF authorization.
- Provides specific NF Type selection based on subscriber identity.
- Supports forwarding of messages from one NRF to another NRF.
- Supports georedundancy to ensure service availability.

The OCNRF interacts with every other NF in the 5G core network and it supports the above functions through the following services:

- Management Service
- Discovery Service
- Access Token Service

CNC Console provides an interface to configure global and service parameters in NRF.

For more information about configuring parameters, see the *Configuring NRF Using CNC Console* section in *Oracle Communications Cloud Native Core Network Repository Function User Guide*.

Network Slice Selection Function (NSSF)

NSSF provides the following functions:

Network slices enable the users to select customized networks with different functionalities (such as mobility) and performance requirements (such as latency, availability and reliability). Network slices differ in features supported and network function optimizations. In such cases, network slices may have different S-NSSAIs with different slice and service types. The user can deploy instances of multiple network slices delivering the same features but for different groups of User Equipments (UEs). These instances deliver different committed services as they are dedicated to a customer. The network slices may have different S-NSSAIs with the same slice or service type but different slice differentiators. The OCNSSF fulfills the requirement for determining the individual network function pertaining to a slice.

OCNSSF is a functional element that supports the following functionalities:

- OCNSSF enables the Access and Mobility Management Function (AMF) to perform initial registration and Protocol Data Unit (PDU) session establishment.
- AMF can retrieve NRF, NSI ID, and target AMFs as part of UE initial registration and PDU establishment procedure.
- OCNSSF uses an NF Service Consumer (AMF) to update the S-NSSAI(s) that AMF supports and notifies of any changes in the status.
- OCNSSF selects the network slicing instance (NSI) and determines the authorized Network Slice Selection Assistance Information (NSSAIs) and AMF to serve the UE.
- OCNSSF interaction with NRF allows retrieving specific NF services to be used for registration request.

OCNSSF provides the following information when queried by the AMF:

- Allowed NSSAIs

- Configured NSSAIs
- Restricted NSSAIs
- Candidate AMF List (in case of registration)
- Network Slice instance ID (for PDU session establishment)
- Slice-level NRF information (for PDU Connectivity)

OCNSSF supports the above functions through the following NSSF services:

- NS Selection service (Nnssf_NSSelection): This service is used by an NF Service Consumer (AMF) to retrieve the information related to network slice. It enables network slice selection in the serving Home Public Land Mobile Network (HPLMN).
- NS Availability Service (Nnssf_NSAvailability): This service stores and maintains list of supported S-NSSAIs per TA. It allows NF service Consumer (AMF) to update and subscribe the above data and get notifications for any addition or deletion of supported S-NSSAIs.

CNC Policy

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

CNC Policy supports the above functions through the following services:

- Session Management Service
- Access and Mobility Service
- Policy Authorization Service
- User Equipment (UE) Policy Service
- PCRF Core Service

CNC Console provides an interface to configure policies and manageable objects in CNC Policy.

For more information about configuring parameters, see the *Configuring CNC Policy Using CNC Console* section in *Oracle Communications Cloud Native Core Policy User Guide*.

Service Communication Proxy (SCP)

SCP provides the following functionalities to other 5G Network Functions (NFs):

- Routing/Selection: Routing rules, refresh cache, and handle application failures and redirects.
 - Dynamic Discovery: The 5G topology is determined from Network Repository Function (NRF) and creation of routing rules.
 - Static Configuration: Enables NF Profiles configuration.
- Load Balancing: Load balancing based on static capacity, NF Type, NF Specific, and NF Priority as mentioned in the NF Profile.
- NF Subscription: Subscription for all NF types.
- Circuit Breaking: Initiated on a per FQDN basis when outstanding transactions exceed a configurable value.
- Message Priority : Message Priority assignment and override based on the 3GPP-SBI-Message-Priority header.
- Congestion and Overload: Uniform load balancing and routing strategy across the network and protects the pod from overload related to various system resources.

CNC Console provides an interface to configure the SCP features.

For more information about configuring parameters, see the *Configuring SCP Using CNC Console* section in *Oracle Communications Cloud Native Core Service Communication Proxy User Guide*.

Security Edge Protection Proxy (SEPP)

OCSEPP supports the following functionalities:

- Protects application layer control plane messages and sensitive data between two NFs belonging to different PLMNs that use the N32 interface to communicate with each other. The N32 interface is used between the SEPPs of a VPLMN and a HPLMN in roaming scenarios. 3GPP has specified N32 to be considered as two separate interfaces: N32-c and N32-f.
 - N32-c is the Control Plane interface between the SEPPs for performing the initial handshake and negotiating the parameters to be applied for the actual N32 message forwarding.
 - N32-f is the Forwarding interface between the SEPPs, that is used for forwarding the communication between the Network Function (NF) service consumer and the NF service producer after applying the application level security protection.
- Provides secure communication of Inter PLMN messages from Consumer NF to Producer NF using TLS protection mode (HTTP over TLS)
- Supports configuration of roaming partner profiles using REST API
- Performs mutual authentication and negotiation of cipher suites with the SEPP in the roaming partner's network

- Handles key management aspects that involve setting up the required cryptographic keys needed for securing messages on the N32 interface between two SEPPs
- Provides a single point of access and control to internal NFs
- Validates inbound traffic as to whether it is from an authorized external PLMN
- Supports cross-layer validation of source and destination addresses and identifiers to provide anti-spoofing capabilities

CNC Console provides an interface to configure different services in SEPP.

For more information about configuring parameters, see the *Configuring SEPP Using CNC Console* section in *Oracle Communications Cloud Native Core Security Edge Protection Proxy User Guide*.

Unified Data Repository (UDR)

Oracle's 5G UDR :

- Leverages a common Oracle Communications Cloud Native Framework
- Is compliant to 3GPP 29.505 Release 15 specification UDM
- Is compliant to 3GPP 29.519 Release 16 (backward compatible with Release 15) specification for PCF
- Has tiered architecture providing separation between the connectivity, business logic, and data layers
- Uses Oracle MySQL NDB Cluster CGE Edition as backend database in the Data Tier
- Registers with NRF in the 5G network so that the other NFs in the network can discover UDR through NRF
- Registers UDR with services like DR-SERVICE and GROUP-ID-MAP

CNC Console provides an interface to configure global and service parameters in UDR.

For more information about configuring parameters, see the *Configuring UDR Using CNC Console* section in *Oracle Communications Cloud Native Core Unified Data Repository User Guide*.

Managing CNC Console Support for NF GUI

Enable and Configure each NF

See *CNC Console Supported NF Configurations* in *Oracle Communications Cloud Native Core Console Installation and Upgrade Guide* for more details about enabling and configuring NFs .

Observe

For information on Metrics and KPIs, see [CNC Console Metrics](#), and [CNC Console KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [CNC Console Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see [CNC Console Logs](#).
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

Support for OCCNE Common Services

CNC Console Common Services GUI provides an option to enable cards (hyperlinks) for OCCNE Common services and OSO services.

CNE Common Services

CNC Console Common Services GUI provides an option to enable cards (hyperlinks) for OCCNE Common services such as Grafana, Kibana, Jaeger, Prometheus and AlertManager.

OSO Common Services

CNC Console Common Services GUI also provides an option to enable cards (hyperlinks) for OSO services such as Prometheus and AlertManager.

Managing Common Service Support (OSO Cards and CNE Cards)

Enable and Configure

For enabling and configuring Common Services , see *CNC Console Common Services GUI* section in *Oracle Communications Cloud Native Core Console Installation and Upgrade Guide*.

Observe

For information on Metrics and KPIs, see [CNC Console Metrics](#), and [CNC Console KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [CNC Console Alerts](#) section for resolution steps.

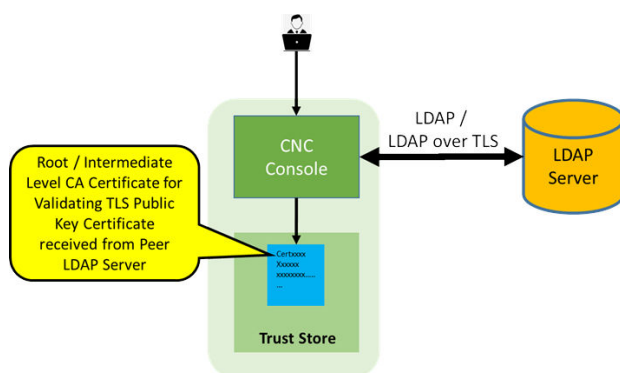
In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see [CNC Console Logs](#).
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

LDAP Integration

The Lightweight Directory Access Protocol (LDAP) is a protocol that defines the technique for accessing the directory data.

Figure 2-1 LDAP



The CNC Console IAM is used as an integration platform to connect it into existing Lightweight Directory Access Protocol (LDAP) and Active Directory (AD) servers.

CNC Console IAM can combine existing external user databases having user and credential details. You can integrate the CNC Console IAM to perform validation of these user credentials and pull in the identity information.

CNC Console IAM provides LDAP over TLS support to securely communicate with external LDAP and active directory servers.

CNC Console IAM also supports Lightweight Directory Access Protocol Secure (LDAPS) between the client and server to make the communication secure.

Managing LDAP Integration

Enable and Configure

For enabling and configuring LDAP integration feature, see [Integrating CNC Console LDAP Server with CNC Console IAM](#)

Observe

For information on Metrics and KPIs, see [CNC Console Metrics](#), and [CNC Console KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [CNC Console Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see [CNC Console Logs](#).
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

SAML 2.0 Integration

SAML (Security Assertion Markup Language) enables applications to authenticate a user using an identity provider. CNC Console can broker identity providers based on the SAML v2.0 protocol.

Managing SAML 2.0 Integration with CNC Console

Enable and Configure

For enabling and configuring SAML 2.0 feature integration with CNC Console, see [Integrating SAML SSO with CNC Console IAM](#) section.

Observe

For information on Metrics and KPIs, see [CNC Console Metrics](#), and [CNC Console KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [CNC Console Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see [CNC Console Logs](#).
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

Logging Support

The CNC Console logs are categorized into following types:

- Regular logs
- Audit logs
- Security logs

Regular logs

These logs contain all kinds of error messages, warnings, or other events written within the application which 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 Keycloak container records the Logging Error Logs as follows:

```
^[[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

These logs contain the header, payload, method, scheme, URI, and other details for all the requests and corresponding responses.

Disabling Security Logs

By default, **Security Log** is enabled for both *CCNC Core* and *CNCC IAM*. To disable, set *securityLogEnabled* flag to **false** in *custom-core_values.yaml* and *custom-iam_values.yaml* files.

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

Log Levels

The log level indicates the level of the logs.

The default log levels for CNC Console Core is as follows:

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

The default log levels for CNC Console IAM is as follows:

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

Managing Security Logs and Audit Logs

Enable and Configure

For enabling and configuring Security Logs and User Activity Logs, see [CNC Console Logs](#) section.

Observe

For information on Metrics and KPIs, see [CNC Console Metrics](#), and [CNC Console KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [CNC Console Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see [CNC Console Logs](#).
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

Support for Multi Cluster Deployment

Multi cluster is a method of deploying an application on or across multiple Kubernetes clusters for improving availability, isolation, and scalability.

Support for Multicloud Deployment for NFs

CNC Console supports NF deployment across Kubernetes clusters using Manager CNC Console IAM (M-CNCC IAM), Manager CNC Console Core (M-CNCC Core), and Agent CNC Console Core (A-CNCC Core). In a multicloud deployment, the CNC Console can manage NFs and OCCNE common services deployed in the remote Kubernetes clusters.

Support for Multiple Instances of NFs within a cluster

CNC Console supports multiple instances of NFs within a Kubernetes cluster using Manager CNC Console IAM (M-CNCC IAM), Manager CNC Console Core (M-CNCC Core), and Agent CNC Console Core (A-CNCC Core).

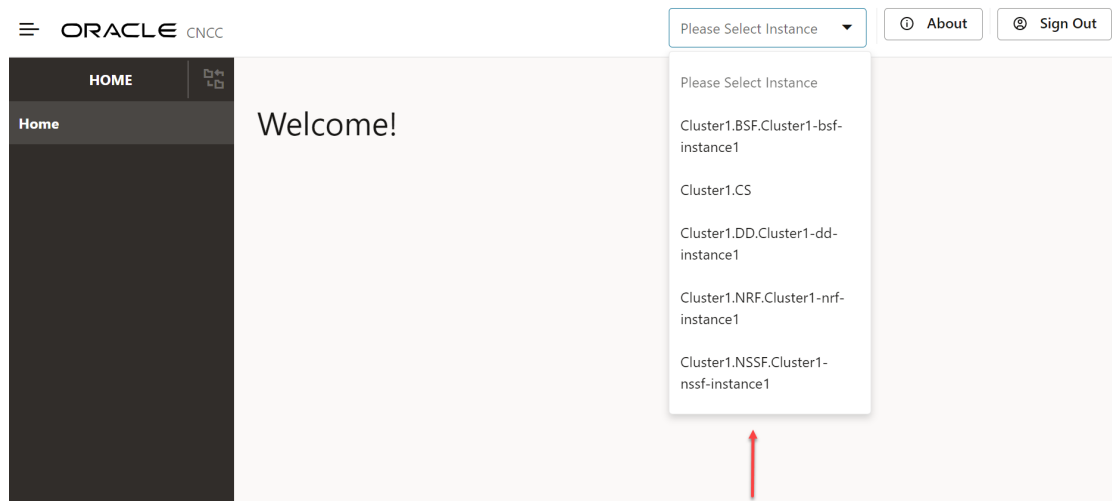
The support for multi cluster deployment mTLS Configuration is added to provide secure communication between CNC Console Master and Agent.

Selecting the Instance

CNC Console multi cluster deployment has introduced a drop-down on header pane for selecting instance.

Values configured in M-CNCC Core instances section gets displayed in the drop down. Naming convention used for instance drop down display is <owner>.<type>.<instance id>

Figure 2-2 Selecting the Instance



Once the user selects the required instance from the drop-down, then the corresponding menu gets loaded.

Figure 2-3 Loading the Menu



The common service instances configured in the instances section gets displayed in the drop down.



For more details about multi-cluster configurations, see *CNC Console Multi-Cluster Configurations* section in *Oracle Communications Cloud Native Core Console Installation and Upgrade Guide*.

cnDBTier Integration

CNC Console is integrated with cnDBTier in a containerized Oracle Communications Cloud Native Environment. For more information, see *Oracle Communications cnDBTier Installation Guide*.

TLS Certificate Support

Transport Layer Security (TLS) certificates or Secure Sockets Layer (SSL) are essential to securing internet browser connections and transactions through data encryption. TLS/SSL is the standard security technology that works behind the scenes to keep the online transactions and logins secure. HTTPS enables end to end encryption of messages to ensure security of data. HTTPS requires creation of TLS (Mutual TLS by two way exchange of ciphered keys).

This feature enables the extension of identity validation from Transport layer to the Application layer. It also provides a mechanism to validate the NF FQDN presence in TLS certificate as added by the Service Mesh against the NF Profile FQDN present in the request.

HTTPS requires creation of TLS (Mutual TLS by two way exchange of ciphered keys).

Steps to Enable HTTPS

Certificate Creation

To create certificate you must have the following files:

- ECDSA private key and CA signed certificate of OCNRF (if initial algorithm is ES256)
- RSA private key and CA signed certificate of OCNRF (if initial algorithm is RSA256)
- TrustStore password file
- KeyStore password file
- CA certificate

Secret Creation

Run the following command to create a secret:

```
$ kubectl create secret generic occnccaccesstoken-secret --from-file=ecdsa_private_key_pkcs8.pem --from-file=rsa_private_key_pkcs1.pem --from-file=trustStorePassword.txt --from-file=keyStorePassword.txt --from-file=ecdsa_ocncc_certificate.crt--from-file=rsa_ocncc_certificate.crt -n occncc
```

Certificate and Key Exchange

Once the connection is established between the client and the server, using the agreed algorithm and keys, they can send messages securely to each other. The TLS handshake has the following phases:

- Hello
- Certificate Exchange
- Key Exchange

The client generates a random key to be used for the main, symmetric algorithm. The client encrypts the key using an algorithm also agreed upon the server's public key (found on its SSL certificate). The client sends this encrypted key to the server, where it is decrypted using the server's private key. The parties are identified that they are talking to the right person, and have secretly agreed on a key to symmetrically encrypt the data that they are about to send each other. The HTTP requests and responses can be sent by forming a plain text message and then encrypting and sending it. The other party is the only one who knows how to decrypt this message, and so Man In The Middle Attackers are unable to read or modify any requests that they may intercept.

CNC Console supports the following cipher suites:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Service Mesh Communication

CNC Console leverages the Istio or Envoy service mesh (Aspen Service Mesh) for all internal and external communication. The service mesh integration provides Console-NF communication and allows API gateway co-working with service mesh. The service mesh integration supports the services by deploying a special sidecar proxy in the environment to intercept all network communication between microservices.

For more information, see *CNC Console Configuration to Support ASM and OSO* in *Oracle Communications Cloud Native Core Console Installation and Upgrade Guide*.

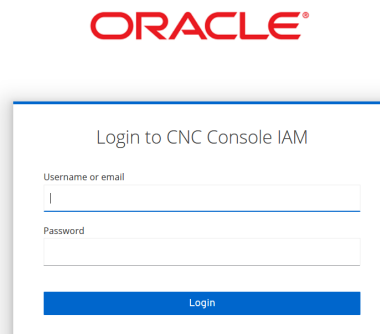
3

Logging into CNC Console

The following is the procedure to log in to CNC Console:

1. Open any browser.
2. Enter the URL: ***http://<host name>:<port number>***.
where, *host name* is cncc-iam-ingress-ip and *port number* is cncc-iam-ingressport.

The following screen appears:



3. Enter the valid credentials.
4. Click **Login**. The Welcome page of CNC Console interface appears.



Note:

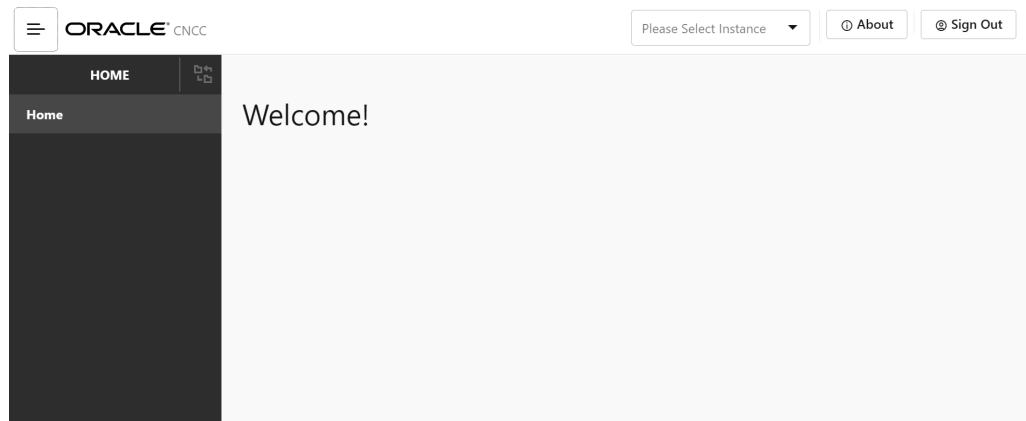
To configure CNCC IAM, see [Configuring CNC Console IAM](#) section.

CNC Console Dashboard

This section provides an overview of the CNC Console Dashboard.

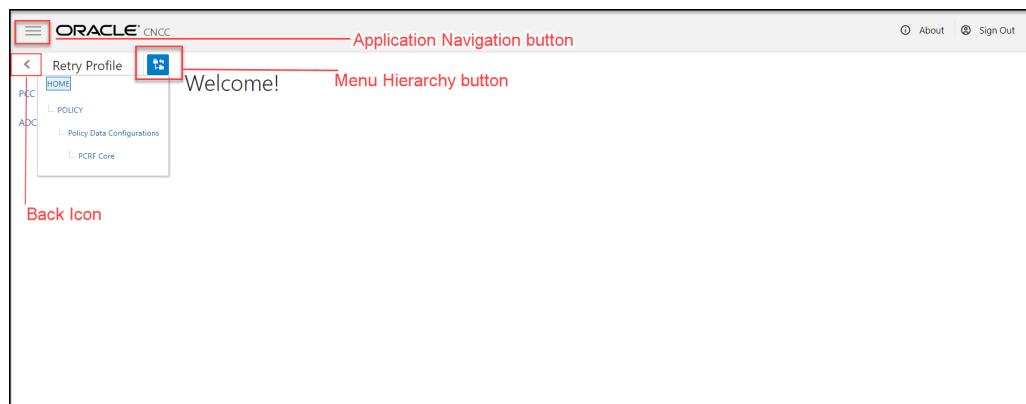
After the user logs in, the following Welcome page of the CNC Console appears:

Figure 3-1 Welcome Page of the CNC Console



- **Top Ribbon** -The top ribbon contains the following:
 - **Instance selection drop-down list**- You can select the NF instance from the list.
 - **About**- Conveys the product name and the version of the Interface.
 - **Sign Out**- To sign out from the Console.
- **Left Pane - NFs and Configurations**
The left pane displays the selected Network Function and the configurations.
- **Right Pane - Details View**
The right pane displays the configurable parameters that can be updated in the selected NFs.
- **Other dashboard options**

Figure 3-2 Dashboard options



- The **Menu Hierarchy** button shows the navigation path from the home screen to the current menu item.
- The **Application Navigation** button allows the user to collapse the left pane and displays full screen.

- The **Back Icon** allows the user to navigate to the Home menu. The screen does not get refreshed automatically. User must click Home or NF menu to view the updated screen.

4

Configuring CNC Console IAM

This section provides details on how to configure the CNC Console IAM.

Role Based Access Control in CNC Console IAM

Role-Based Access Control (RBAC) is one of the main methods for advanced access control. It enables you to restrict network access to authorized users based on their assigned roles.

Role

A Role is a collection of permissions that you can apply to users. Roles are defined according to the authority and responsibility of the users within the organization. Using roles makes it easier to add, remove, and update permissions to the users.

Composite Role

A Composite Role is a collection of one or more additional roles grouped together.

Types of Roles in CNC Console

Role Based Access Control (RBAC) is controlled by Identity and Access Management (IAM) functionality provided by CNCC IAM.

The following roles are predefined in the CNC Console IAM:

- **ADMIN Level**

Role: **ADMIN**

The user assigned with this role can access all the NF and CS resources within the CNC Console. The admin user can perform create, read, update, and delete operations. For example, the admin user can read, add, update, or delete MOS configurations for any NF and CS supported CNC Console application. The admin user can also be assigned to the Composite roles which contain all NF and CS level roles.

- **NF Level**

The user assigned with this role can perform read and write operations for the assigned NFs. NF level roles are classified into:

<NF>_READ: With this permission, the assigned user can perform the read operation for NFs.

For example, If user has POLICY_READ role, then the user can only read configurations of any MOs configurations within the Policy and cannot write or update or delete any record.

<NF>_WRITE: With this permission, the assigned user can perform create, read, update, and delete operations for NFs. For example, if user has POLICY_WRITE then the user can read or write or update or delete any MOs configurations within the NF.

- **CS Level**

Role: **CS_WRITE**

The user assigned with this role has access to all the common services and can perform CRUD operations.

The user can perform create, read, update, and delete operations. The user can read, add, update, or delete MOS configurations for all common services such as Grafana, Kibana, Jaeger, Prometheus, Alertmanager supported by CNC Console application. For example, if user has CS_WRITE then, the user can read or write or update or delete any MOs configurations in common services.

Accessing Roles in CNC Console Applications

1. Log into CNC Console IAM using Admin credentials. The following screen appears.

Figure 4-1 Realm Settings

The screenshot shows the Oracle CNC Console IAM interface. The left sidebar contains a navigation menu with options like Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication, Groups, Users, Sessions, Events, Import, and Export. The 'Realm Settings' option is selected. The main panel displays the 'General' tab of the Realm Settings. The fields are as follows:

- Name: cncc
- Display name: cncc
- HTML Display name: <div class="lc-logo-text"></div>
- Frontend URL: (empty)
- Enabled: ON
- User-Managed Access: OFF
- Endpoints: OpenID Endpoint Configuration, SAML 2.0 Identity Provider Metadata

2. To access or view the available roles, click **Roles** on the left pane. The defined roles are available on the right pane.

Figure 4-2 Roles

The screenshot shows the Oracle CNC Console IAM interface. The left sidebar contains a navigation menu with options like Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication, Groups, Users, Sessions, Events, Import, and Export. The 'Roles' option is selected. The main panel displays the 'Roles' tab with a table of roles.

Role Name	Composite	Description	Actions
ADMIN	True	Has access to all NF resources and can perform CRUD operations	Edit Delete
BSF_READ	False	Has access to only BSF resources and can only perform READ Managed Objects of BSF.	Edit Delete
BSF_WRITE	True	Has access to only BSF resources and can perform CRUD operation on Managed Objects of BSF.	Edit Delete
CS_WRITE	False	Has access to only CS resources and can perform CRUD operations on Managed Objects.	Edit Delete
default-roles-cncc	True	\$(role_default-roles-cncc)	Edit Delete
NRF_READ	False	Has access to only NRF resources and can only perform READ Managed Objects of NRF.	Edit Delete
NRF_WRITE	True	Has access to only NRF resources and can perform CRUD operations on Managed Objects.	Edit Delete
NSSF_READ	False	Has access to only NSSF resources and can only perform READ Managed Objects of NSSF.	Edit Delete
NSSF_WRITE	True	Has access to only NSSF resources and can perform CRUD Managed Objects of NSSF.	Edit Delete
offline_access	False	\$(role_offline-access)	Edit Delete
POLICY_READ	False	Has access to only POLICY resources and can only perform READ Managed Objects of POLICY.	Edit Delete
POLICY_WRITE	True	Has access to only POLICY resources and can perform CRUD operation on Managed Objects of POLICY.	Edit Delete
SCP_READ	False	Has access to only SCP resources and can only perform READ Managed Objects of SCP.	Edit Delete
SCP_WRITE	True	Has access to only SCP resources and can perform CRUD operations on Managed Objects.	Edit Delete
SEPP_READ	False	Has access to only SEPP resources and can only perform READ Managed Objects of SEPP.	Edit Delete
SEPP_WRITE	True	Has access to only SEPP resources and can perform CRUD operations on Managed Objects of SEPP.	Edit Delete
UDR_READ	False	Has access to only UDR resources and can only perform READ Managed Objects of UDR.	Edit Delete
UDR_WRITE	True	Has access to only UDR resources and can perform CRUD Managed Objects of UDR.	Edit Delete
uma_authorization	False	\$(role_uma_authorization)	Edit Delete



Note:

For the details about Multi-Cluster Roles, see **Multi-Cluster Roles** under **CNC Console IAM Post Installation Steps** section in *Oracle Communication Cloud Native Core Console Installation and Upgrade Guide*.

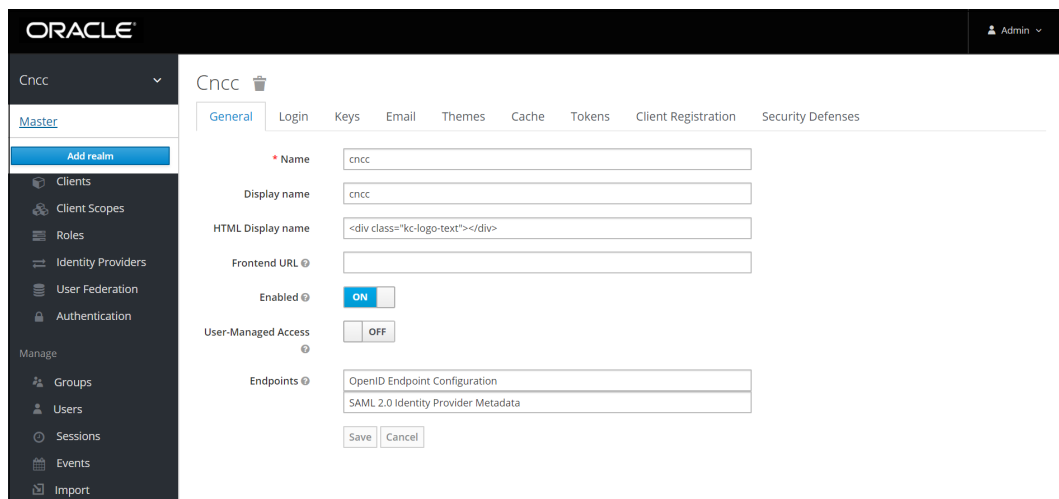
Creating or Updating Admin Password in CNC Console IAM

This section describes about creating or updating the admin password in CNC Console IAM.

Perform the following steps to create or update the admin password:

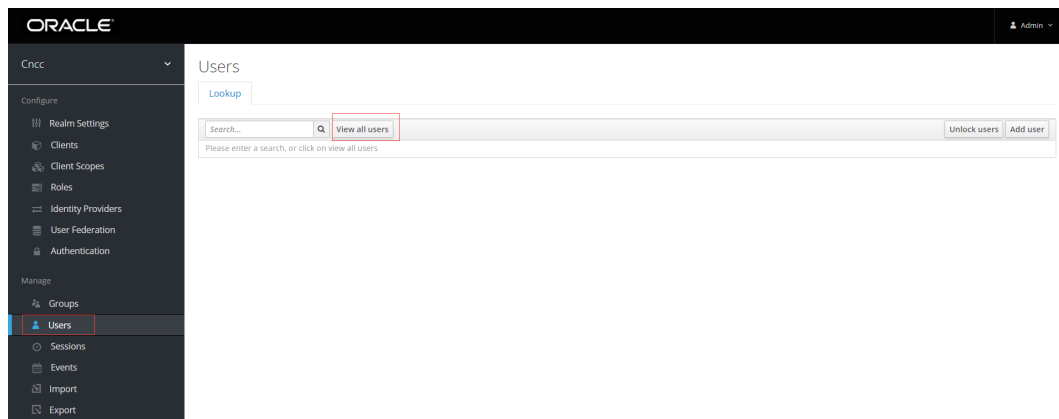
1. Select the **Master Realm** (Only applicable for Admin Users).

Figure 4-3 Master Realm



2. Click **Users** on the left pane and click **View all users** on the right pane.

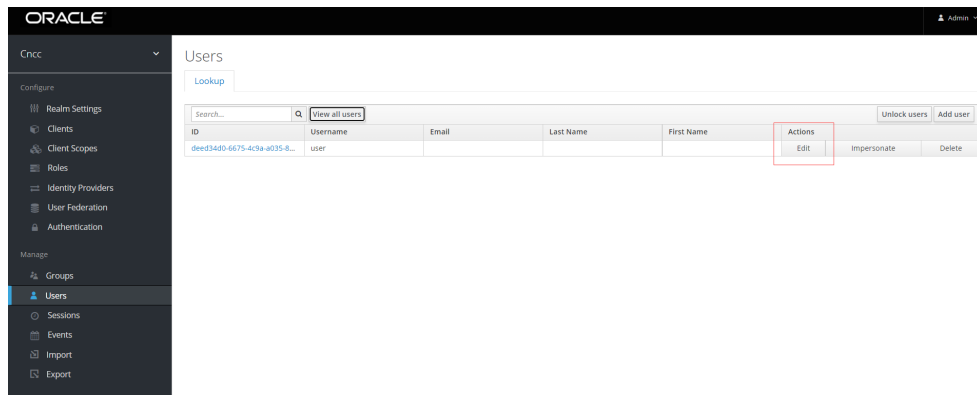
Figure 4-4 view all users



The following screen appears.

3. Click **Edit** under **Actions** to update the credentials.

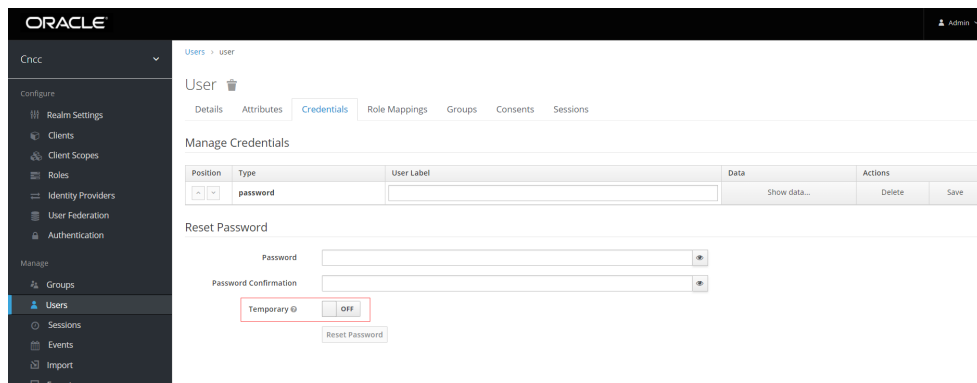
Figure 4-5 Edit User



The following screen appears.

4. Under **Credentials** tab, set **Temporary** to **OFF** and update the **Password**.

Figure 4-6 Credentials



Creating or Updating User Password in CNC Console IAM

This section describes about creating or updating the user password in CNC Console IAM.

Perform the following steps to create or update the user password:

1. On the left pane, click **Realm Settings**. The following screen appears.

Figure 4-7 Realm Settings

The screenshot shows the Oracle CNC Console IAM interface. The left sidebar is under the 'Configure' section, with 'Realm Settings' selected. The main content area is the 'General' tab of the 'Cncc' realm configuration. It includes fields for Name, Display name, HTML Display name, Frontend URL, and toggle switches for Enabled (ON) and User-Managed Access (OFF). There are also fields for Endpoints (OpenID Endpoint Configuration and SAML 2.0 Identity Provider Metadata) and 'Save' and 'Cancel' buttons.

2. Click **Users** on the left pane and click **view all users** on the right pane.

Figure 4-8 view all users

The screenshot shows the Oracle CNC Console IAM interface. The left sidebar is under the 'Manage' section, with 'Users' selected. The main content area is the 'Users' section, which has a 'Lookup' tab and a search bar. A 'View all users' button is highlighted in the search bar area. There are also 'Unlock users' and 'Add user' buttons.

The following screen appears.

3. Click **Edit** under **Actions** to update the credentials.

Figure 4-9 Users

The screenshot shows the Oracle CNC Console IAM interface. The left sidebar is under the 'Manage' section, with 'Users' selected. The main content area is the 'Users' section, which has a 'Lookup' tab and a search bar. Below the search bar is a table of users. The first user is highlighted, and the 'Actions' column for that user is highlighted, showing 'Edit', 'Impersonate', and 'Delete' options.

ID	Username	Email	Last Name	First Name	Actions
dee03400-6675-4c9a-a035-8...	user				Edit Impersonate Delete

The following screen appears.

- Under **Credentials** tab, set **Temporary** to **OFF** and update the **Password**.

Figure 4-10 Credentials

The screenshot shows the Oracle Identity Cloud Service (OICS) console. On the left, the 'Users' menu is expanded, and the 'Users' option is selected. The main panel displays the 'User' details for a user named 'user'. The 'Credentials' tab is active, showing the 'Manage Credentials' section. A table lists the user's credentials, with one entry of type 'password'. Below the table, the 'Reset Password' section is visible, showing the 'Temporary' checkbox is set to 'OFF' (highlighted with a red box). The 'Reset Password' button is also visible.

Configuring the CNC Console Redirection URL

After a successful deployment of CNC Console IAM, the administrator must perform the following steps to configure the CNC Console redirection URL:

- Log into CNC Console IAM using admin credentials provided during installation.
- On the left pane, select **Clients** and on the right pane select **cncc**.

Figure 4-11 Clients Screen

The screenshot shows the Oracle Identity Cloud Service (OICS) console. On the left, the 'Clients' menu is expanded, and the 'Clients' option is selected. The main panel displays the 'Clients' screen with a search bar and a table of clients. The 'cncc' client is highlighted in red in the table.

Client ID	Enabled	Base URL	Actions
account	True	http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/cncc/auth/realms/cncc/account/	Edit Export Delete
admin-cli	True	Not defined	Edit Export Delete
broker	True	Not defined	Edit Export Delete
cncc	True	Not defined	Edit Export Delete
realm-management	True	Not defined	Edit Export Delete
security-admin-console	True	http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/cncc/auth/admin/cncc/console/	Edit Export Delete

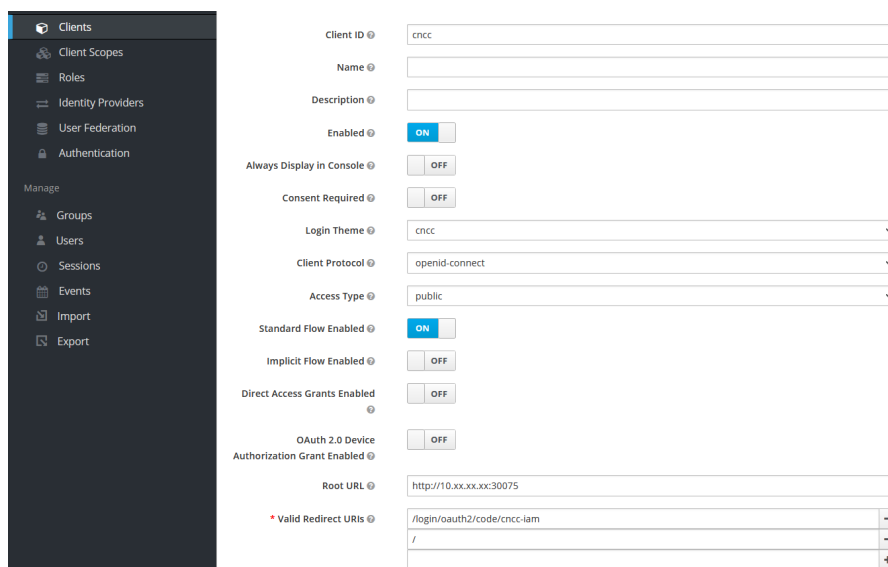
- Enter CNCC Core Ingress URI in the **Root URIs** field and **Save**.

<scheme>://<cncc-mcore-ingress IP/FQDN>:<cncc-core-ingress Port>

 Note:

Redirection url is pre-populated, only root url needs to be configured as part of Post-Installation procedure.

Figure 4-12 Redirection url



The screenshot shows the 'Clients' configuration page in the Oracle Identity Cloud Service console. The left sidebar contains the following menu items: Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication, Manage, Groups, Users, Sessions, Events, Import, and Export. The main configuration area for the client 'cncc' includes the following fields and controls:

- Client ID: cncc
- Name: (empty)
- Description: (empty)
- Enabled: ☒ ON
- Always Display in Console: ☐ OFF
- Consent Required: ☐ OFF
- Login Theme: cncc
- Client Protocol: openid-connect
- Access Type: public
- Standard Flow Enabled: ☒ ON
- Implicit Flow Enabled: ☐ OFF
- Direct Access Grants Enabled: ☐ OFF
- OAuth 2.0 Device Authorization Grant Enabled: ☐ OFF
- Root URL: http://10.xx.xx.xx:30075
- Valid Redirect URIs: /login/oauth2/code/cncc-iam

Users in CNC Console IAM

This section includes:

- [Creating the users](#)
- [Viewing the users](#)
- [Assigning the roles to the users](#)

 Note:

For the details about Setting or updating the admin password refer [Creating or Updating Admin Password in CNC Console IAM](#)

 Note:

For the details about Setting or updating the user password refer [Creating or Updating User Password in CNC Console IAM](#)

Creating the Users

1. Click **Realm Settings** under **Cncc**.

2. Click **Users** under **Manage** on the left pane and click **Add user** on the right pane.

3. **Add user** screen appears. Add the user details and click **Save**.

4. The user has been created and the user details screen appears.

- For setting the password for the user, click **Users** under **Manage** on the left pane and click the **Credentials** tab on the right pane. Set the password for the user.

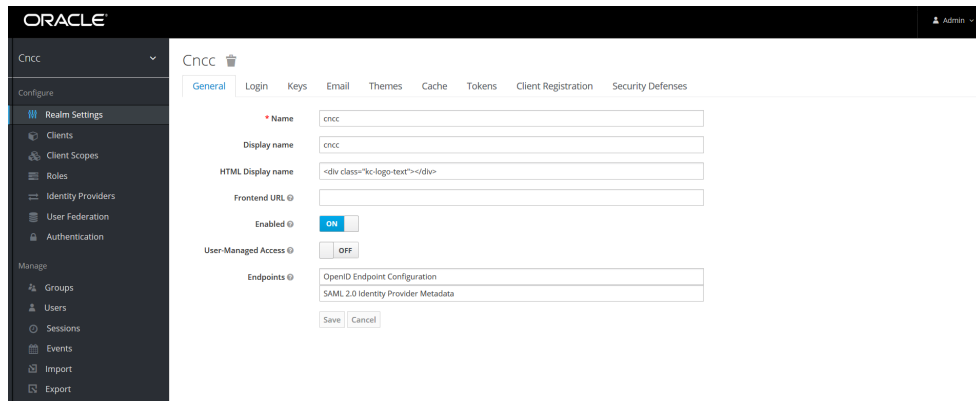


Note:

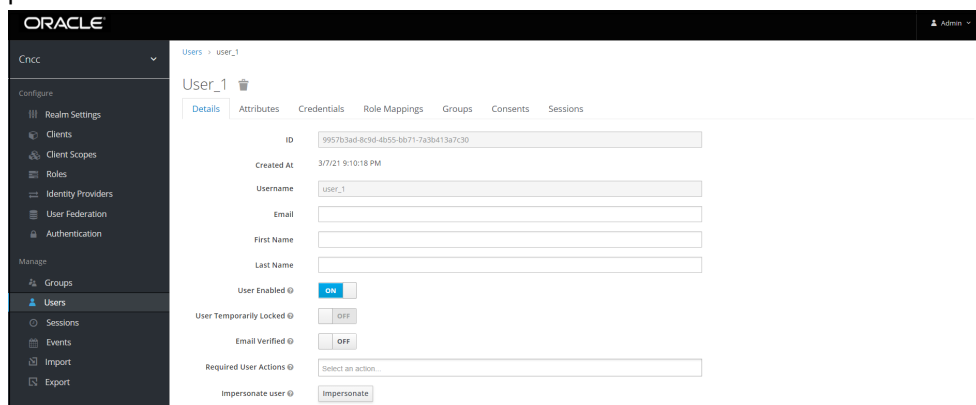
Setting the **Temporary** flag **ON** prompts the user change the password when logging in to the CNC Console for the first time.

Viewing the Users

- Click **Realm settings**.



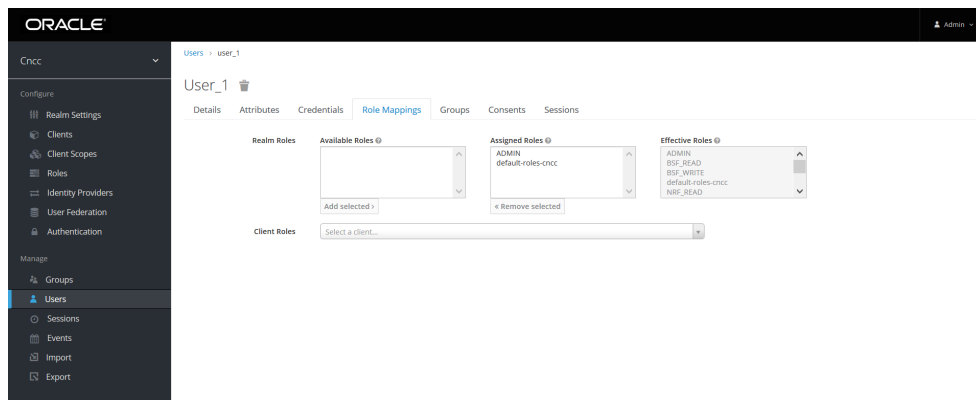
2. Select **Users** under **Manage** in the left pane and select **View all users** in the right pane.



The list of users and their details appears in the right pane.

Assigning Roles to the User

1. Select **Users** under **Manage** in the left pane and click **View all users** in the right pane. Choose any user. Select **Role Mappings** tab in the right pane of the user screen and select the roles from **Available roles** and click **Add selected**.



The selected roles will be assigned to the user.

Integrating SAML SSO with CNC Console IAM

Overview

Security Assertion Markup Language (SAML) is an open standard that allows identity providers (IdP) to pass authorization credentials to service providers (SP). The identity provider authenticates the user and returns the assertion information about the authenticated user and the authentication event to the application. Using SSO, if the user tries to access any other application that uses the same identity provider for user authentication, the user need not login again. This is the principle of SSO (Single Sign On).



Note:

CNCC supports SAML 2.0.

Configuring SAML Identity Provider in CNCC IAM

1. To configure SAML identity provider (**IdP**) in CNCC IAM, log in to CNCC IAM Console using admin credentials provided during installation of CNCC IAM .

`http://<cncc-iam-ingress-external-ip>:<cncc-iam-ingress-service-port>`
Example: `http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/`

Figure 4-13 Login screen

ORACLE®

Login to CNC Console IAM

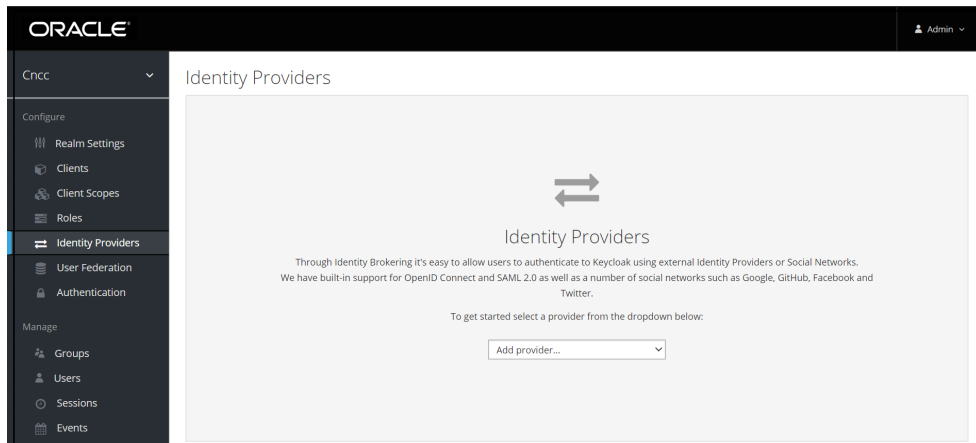
Username or email
|

Password

Login

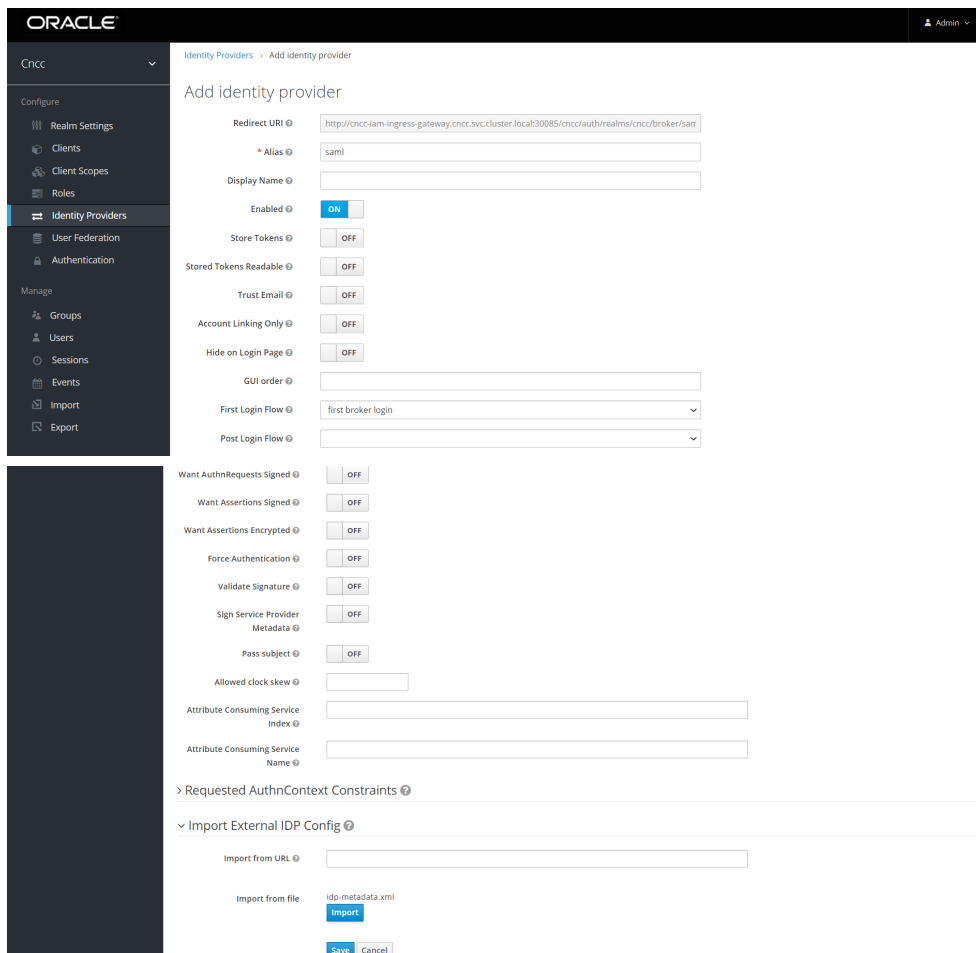
2. Click **Cncc** realm and click the **Identity Provider** tab on the left pane. **Identity Providers** screen appears on the right pane.

Figure 4-14 Identity Provider Screen



3. Choose the **SAML v2.0** from the **Add provider** drop-down list. The **Add Identity Provider** screen appears.

Figure 4-15 Add Identity Provider screen



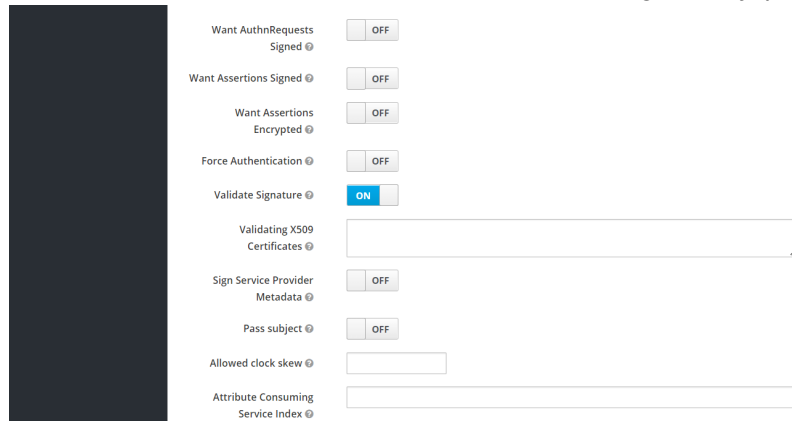
 Note:

- Give an appropriate name for the field **Display Name**.
- At **Import External IDP Config**, upload the 'idp-metadata.xml' file that is exported from SAML client in the IdP.

Click **Import** and **Save**. The other required fields will be filled in automatically.

Perform the following procedure to configure the IDP manually, if you are facing difficulty in importing the metadata file from the IDP Client:

- Select **SAML v2.0** in add Provider selection box
- Set the value of **Single Sign-On Service URL** to the URL of the preferred IDP.
- Example: <IP/FQDN>:<PORT>/auth/realms/master/protocol/saml (URI for their preferred IDP where SAML AuthnRequest will be sent).
- Set the value of **Single Logout Service URL**.
Example: <IP/FQDN>:<PORT>/auth/realms/master/protocol/saml (URI for their preferred IDP where Logout requests must be sent).
- If the IDP supports *HTTP POST* binding methods, enable **HTTP-POST Binding Response**, **HTTP-POST Binding Logout** and **HTTP-POST Binding for AuthnRequest** flags. (By default, HTTP-Redirect will be used)
- If the IDP is sending signed Assertions, set **Want Assertions Signed** to **ON**.
- Set **Validate Signature** to **ON**.
- Provide value for **Validating X509 Certificates** (If you are using Keycloak as an IDP, use the certificate from **master** realm -> Realm Settings -> Keys).

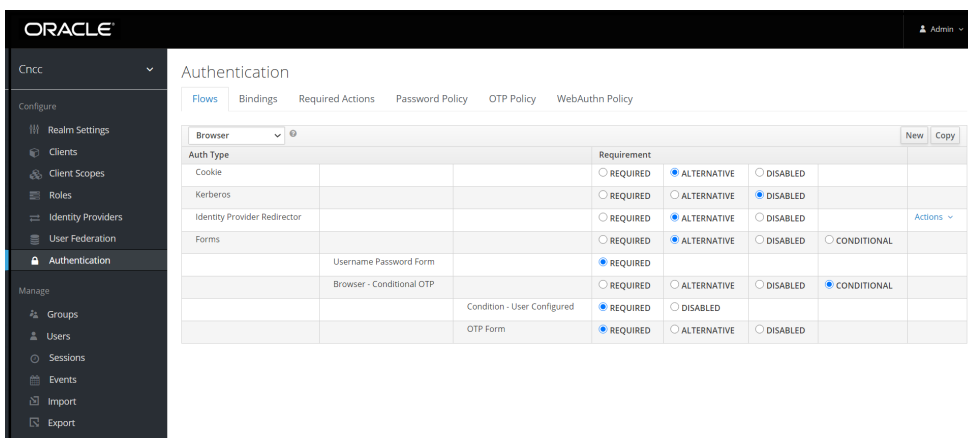


The screenshot shows a configuration form for an IDP. The form includes several toggle switches and text input fields. The 'Validate Signature' toggle is set to 'ON'. The 'Validating X509 Certificates' field is empty. Other toggles include 'Want AuthnRequests Signed', 'Want Assertions Signed', 'Want Assertions Encrypted', 'Force Authentication', 'Sign Service Provider Metadata', 'Pass subject', and 'Allowed clock skew'. The 'Attribute Consuming Service Index' field is also empty.

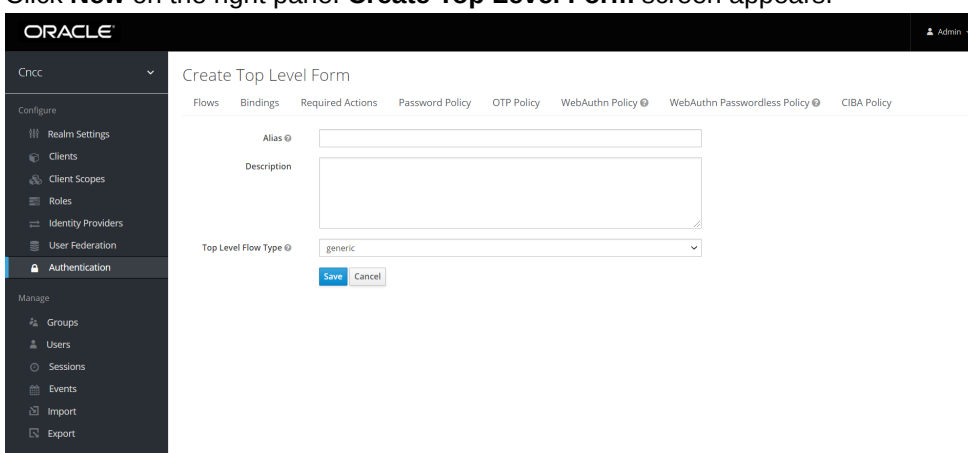
- Click **Save**.

IDP is now configured manually.

- To create custom '**First Login Flow**', click **Authentication** tab on the left pane. The **Authentication** screen appears.

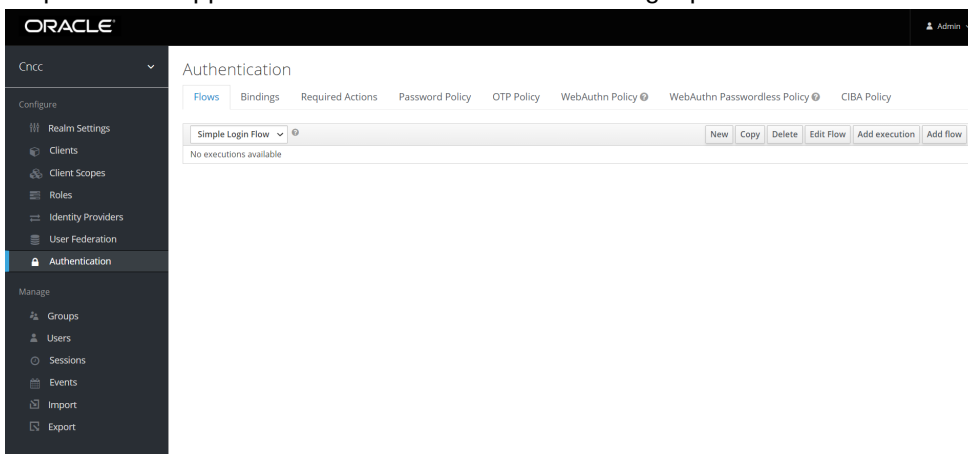


5. Click **New** on the right pane. **Create Top Level Form** screen appears.

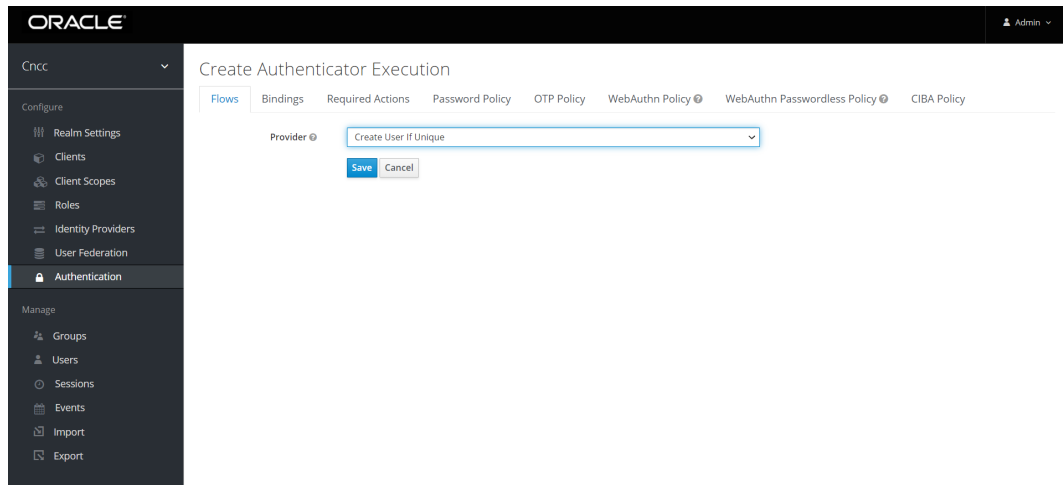


Enter the appropriate alias and click **Save**.

6. The **Authentication** screen with the newly created custom flow selected in the drop-down list appears. Click **Add Execution** on the right pane .

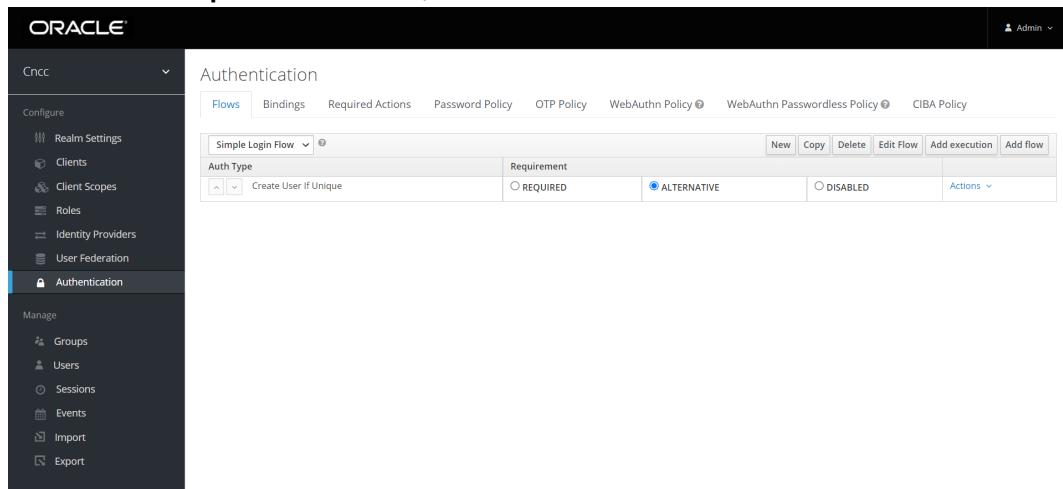


7. **Create Authenticator Execution** screen appears.

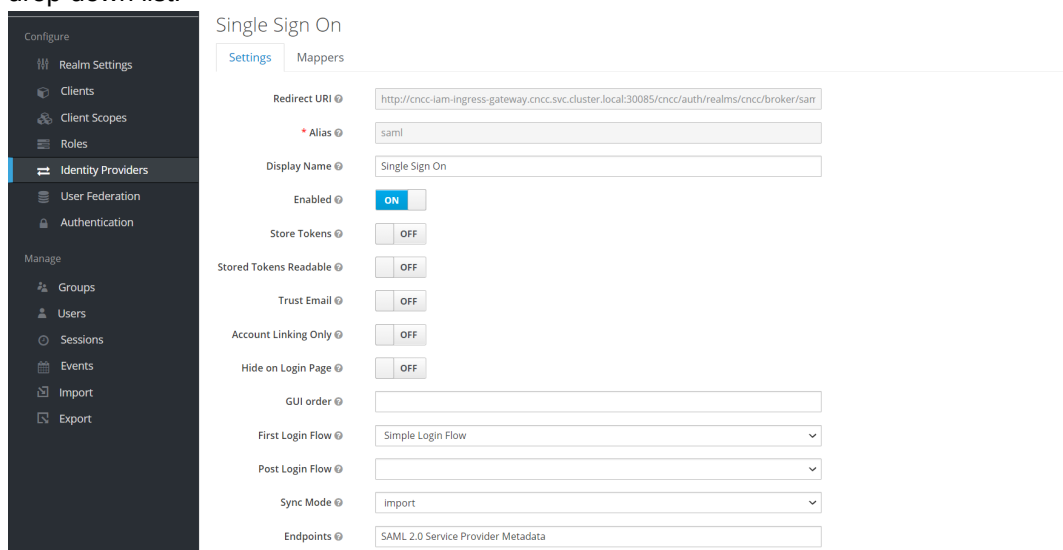


Select **Create User If Unique** from the **Provider** drop-down list. Click **Save**.

8. The **Authentication** screen with the newly created custom flow selected in the drop-down. Under **Requirement** section, select **Alternative**.



9. Click **Identity Provider** in the left pane. Select the custom flow from **First Login Flow** drop-down list.



The above screen appears. Now the SAML Idp roles must be mapped with CNC Console IAM API roles.

**Note:****CNCC IAM(SP) Configuration in IDP**

In a SAML based SSO Implementation, the IDP needs to send SAML assertions towards a Service Provider (CNCC IAM in this case) endpoint.

Use the following CNCC endpoint in the IDP :

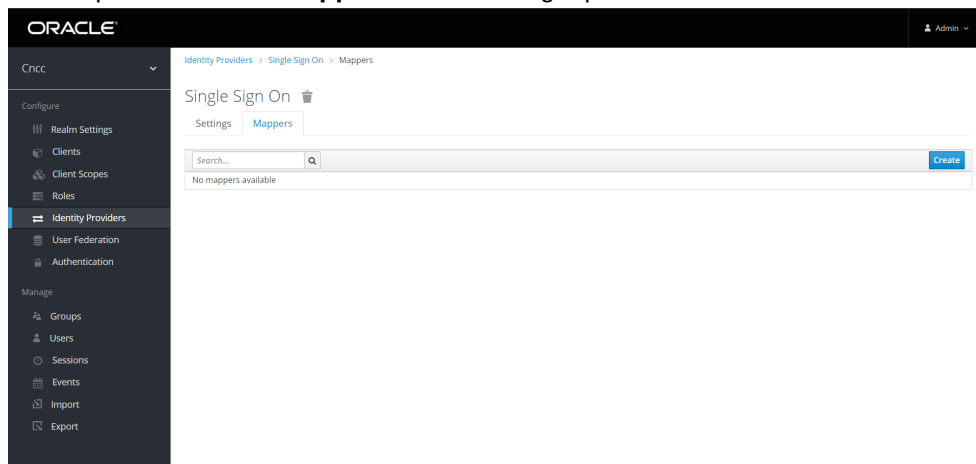
```
http://<IP/FQDN>:<PORT>/cncc/auth/realms/cncc/broker/saml/  
endpoint
```

Example:

```
http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/  
cncc/auth/realms/cncc/broker/saml/endpoint
```

Mapping SAML IdP roles with CNC Console IAM API roles

1. After saving SAML IdP configurations in CNCC IAM, select **Identity Providers** on the left pane and click **Mappers** tab on the right pane. Click **Create**.



2. Add Identity Provider Mapper Screen appears.

ORACLE

Cncc

Identity Providers > saml > Identity Provider Mappers > Create Identity Provider Mapper

Add Identity Provider Mapper

Name *

Sync Mode Override *

Mapper Type

Attribute Name

Friendly Name

Attribute Value

Role

- Give an appropriate name for the field **Identity Provider Mapper**.
- Select 'SAML Attribute to Role' from **Mapper Type** drop-down.
- Enter the **Attribute Value** as the one of the roles added in SAML IdP. Example: 'NRF', 'SCP', etc.
- Click **Select Role** to select the API roles to be enabled for this mapping.
- Click **Save**.

ORACLE

Cncc

Identity Providers > Single Sign On > Mappers

Single Sign On

Settings **Mappers**

Name	Category	Type
nrRole	Role Mapper	SAML Attribute to Role
scpRole	Role Mapper	SAML Attribute to Role

You can create any number of mapping as per the requirements.

Accessing CNCC Core Application

1. To log in to CNCC Core, browse to the application using hostname and port. The user will be redirected to CNCC IAM (broker).

http://<cncc-core-ingress-extrenal-ip>:<cncc-iam-ingress-service-port>
Example: http://cncc-core-ingress-gateway.cncc.svc.cluster.local:30075/

A login form titled "Login to CNCC". It contains two input fields: "Username or email" and "Password". Below these fields is a blue "Log In" button. Underneath the button is a horizontal line, followed by the text "Or sign in with", and then a button labeled "Single Sign On".

2. Click **Single Sign On** to authenticate using SAML SSO. The user is redirected to SAML IdP log in. Enter user details to access CNCC Core application.

Integrating CNC Console LDAP Server with CNC Console IAM

Overview

The CNC Console IAM can be used as an integration platform to connect it into existing LDAP and Active Directory servers.

User Federation in CNC Console-IAM let the user to sync users and groups from LDAP and Active Directory servers and assign roles respectively.

CNCC IAM provides an option to configure a secured connection URL to your LDAP store.

example: ``ldaps://myhost.com:636'`

CNCC IAM uses SSL for communication with the LDAP server. The truststore must be properly configured on the CNCC IAM server side, otherwise CNCC IAM cannot trust the SSL connection to LDAP.

Sample LDAP Idif File

```
dn: dc=oracle,dc=org
objectclass: top
objectclass: domain
objectclass: extensibleObject
dc: oracle

dn: ou=groups,dc=oracle,dc=org
objectclass: top
objectclass: organizationalUnit
ou: groups

dn: ou=people,dc=oracle,dc=org
objectclass: top
objectclass: organizationalUnit
ou: people

dn: uid=ben,ou=people,dc=oracle,dc=org
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Ben Alex
sn: Alex
uid: ben
userPassword: benspass

dn: uid=bob,ou=people,dc=oracle,dc=org
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Bob Hamilton
sn: Hamilton
uid: bob
userPassword: bobspass

dn: uid=joe,ou=people,dc=oracle,dc=org
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Joe Smeth
sn: Smeth
uid: joe
userPassword: joespass

dn: cn=admin,ou=groups,dc=oracle,dc=org
objectclass: top
objectclass: groupOfUniqueNames
cn: admin
uniqueMember: uid=ben,ou=people,dc=oracle,dc=org
ou: admins
```

```
dn: cn=scp,ou=groups,dc=oracle,dc=org
objectclass: top
objectclass: groupOfUniqueNames
cn: scp
uniqueMember: uid=ben,ou=people,dc=oracle,dc=org
uniqueMember: uid=joe,ou=people,dc=oracle,dc=org
ou: scpusers
```

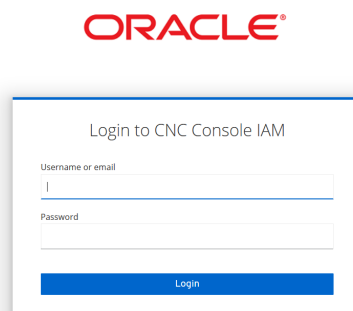
```
dn: cn=nrf,ou=groups,dc=oracle,dc=org
objectclass: top
objectclass: groupOfUniqueNames
cn: nrf
uniqueMember: uid=ben,ou=people,dc=oracle,dc=org
uniqueMember: uid=bob,ou=people,dc=oracle,dc=org
ou: nrfusers
```

Configuring User Federation with CNC Console IAM

This section provides information about configuring user federation with CNC Console IAM (LDAP Server integration).

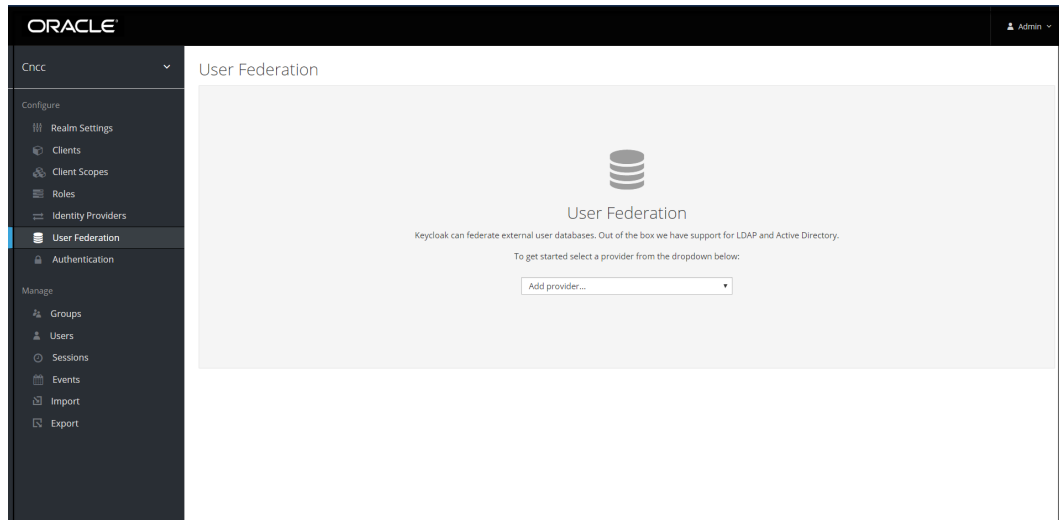
1. Login to CNCC IAM console **`http://<cncc-iam-ingress-ip>:<cncc-iam-ingress-port>`** using admin credentials provided during installation of CNCC IAM.

Figure 4-16 Login Screen



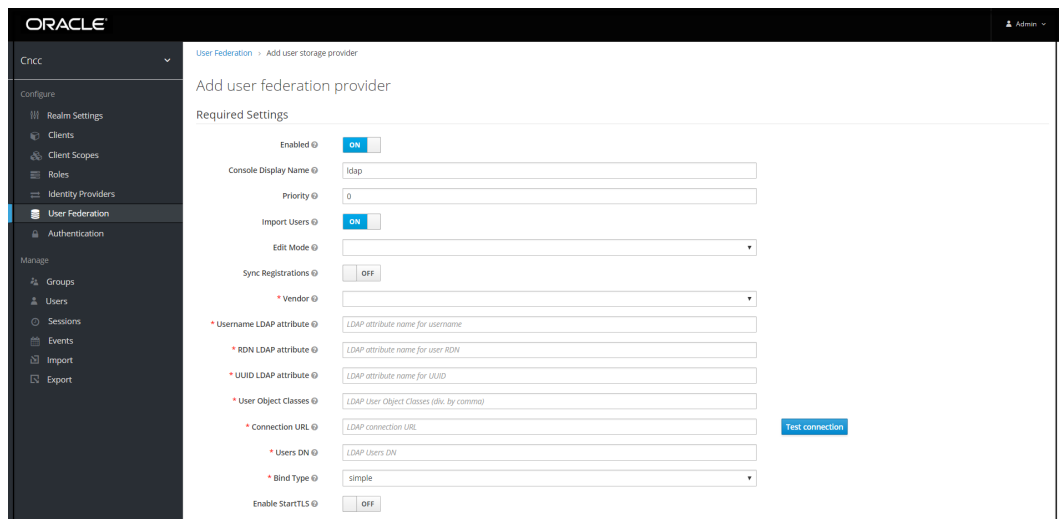
2. Click **Realm Settings** and click **Add realm** under **Cncc**. Click the **User Federation** on the left pane. The **User Federation** screen appears in the right pane.

Figure 4-17 User Federation Screen



- From the drop-down list in the **User federation** screen select **Idap**, the **Add user federation provider** screen appears.

Figure 4-18 Add user federation provider screen

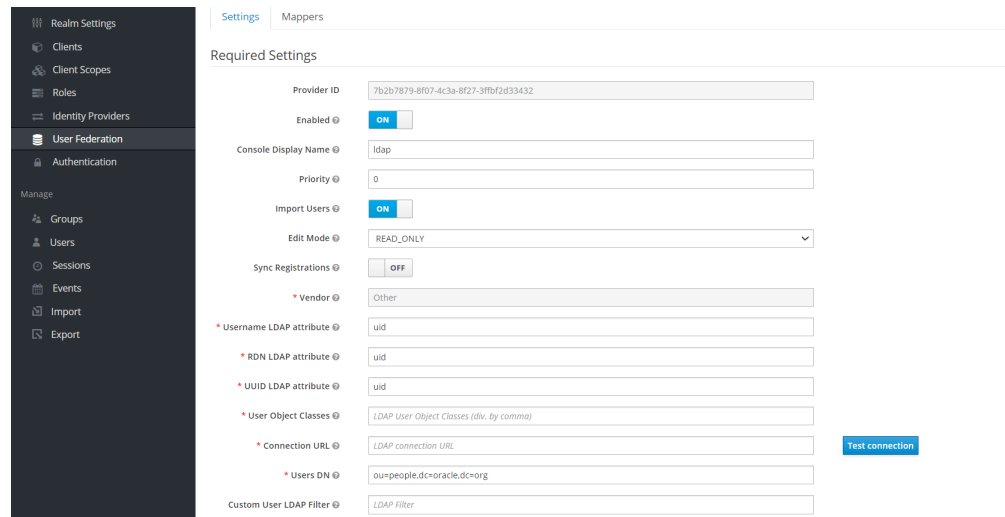


- Enter the values for the following fields:
 - Console Display Name:** Enter the display name.
 - Edit Mode:** Select **READ_ONLY**.
 - Vendor:** Enter the LDAP server provider name for the company.

 Note:

This usually fill the defaults for many of the fields. But in case user have a different setup than the defaults, the correct values must be provided. Current set up is **Spring embedded LDAP**, so select the last option "Other" from the drop-down list.

Figure 4-19 User Federation



The screenshot displays the 'User Federation' configuration interface. On the left is a dark sidebar with a menu containing 'Realms Settings', 'Clients', 'Client Scopes', 'Roles', 'Identity Providers', 'User Federation' (selected), and 'Authentication'. Below this is a 'Manage' section with 'Groups', 'Users', 'Sessions', 'Events', 'Import', and 'Export'. The main content area has tabs for 'Settings' and 'Mappers'. Under 'Settings', the 'Required Settings' section includes the following fields:

- Provider ID**: Text input with value '7b2b7879-8f07-4c3a-8f27-3f5f2d33432'.
- Enabled**: Toggle switch set to 'ON'.
- Console Display Name**: Text input with value 'ldap'.
- Priority**: Text input with value '0'.
- Import Users**: Toggle switch set to 'ON'.
- Edit Mode**: Dropdown menu set to 'READ_ONLY'.
- Sync Registrations**: Toggle switch set to 'OFF'.
- Vendor**: Text input with value 'Other'.
- * Username LDAP attribute**: Text input with value 'uid'.
- * RDN LDAP attribute**: Text input with value 'uid'.
- * UUID LDAP attribute**: Text input with value 'uid'.
- * User Object Classes**: Text input with value 'LDAP User Object Classes (div. by comma)'.
- * Connection URL**: Text input with value 'LDAP connection URL'.
- * Users DN**: Text input with value 'ou=people,dc=oracle,dc=org'.
- Custom User LDAP Filter**: Text input with value 'LDAP Filter'.

A 'Test connection' button is located to the right of the 'Connection URL' field.

- In most case the **UUID LDAP attribute** value set as "entryUUID". If you do not have a suitable value, use an alternate unique identifier.
 - The default setting for **Import Users** is 'ON'. Change it to 'OFF' to disable user sync.
 - Provide company LDAP server details. The same as provided for *ldap-ldif* file, that is, the connection url (hostname prefixed with ldap:// or when LDAP Secure connection enabled (LDAPS) hostname prefix should be ldaps://), the port.
 - If the LDAP is secured then select 'simple' from the **Bind Type** drop-down and provide the admin bind username and password else select **Bind Type** as "none". Sample data for the field **Bind DN** "cn=admin,dc=oracle,dc=org".
 - Click "Test Connection" and "Test Authentication".
 - Set Cache policy as "NO_CACHE".
5. After filling the required fields, the screen appears as below. Click **Save**.

Figure 4-20 User Federation

The screenshot shows the 'User Federation' configuration page. On the left is a sidebar with navigation links: User Federation, Authentication, Manage, Groups, Users, Sessions, Events, Import, and Export. The main content area contains the following fields and controls:

- Console Display Name: ldap
- Priority: 0
- Import Users: ON (toggle)
- Edit Mode: READ_ONLY (dropdown)
- Sync Registrations: OFF (toggle)
- Vendor: Other
- Username LDAP attribute: uid
- RDN LDAP attribute: uid
- UUID LDAP attribute: uid
- User Object Classes: inetOrgPerson, organizationalPerson, person, top
- Connection URL: ldaps://10.75.175.199:30899
- Users DN: ou=people,dc=oracle,dc=org
- Custom User LDAP Filter: LDAP Filter
- Search Scope: One Level (dropdown)
- Bind Type: simple (dropdown)
- Bind DN: cn=admin,dc=oracle,dc=org
- Bind Credential: [masked]

Buttons: 'Test connection' and 'Test authentication' are located on the right side of the form.

6. New buttons (**Synchronize changed users**, **Synchronize all users**, **Remove imported**, **Unlink users**) appears next to the **Save** and **Cancel**.

Figure 4-21 User Federation

The screenshot shows the 'User Federation' configuration page with advanced settings. The sections and their values are:

- Use Truststore SPI: Only for Idaps
- Connection Pooling: ON (toggle)
- Connection Timeout: Connection Timeout
- Read Timeout: Read Timeout
- Pagination: ON (toggle)
- Kerberos Integration:
 - Allow Kerberos authentication: OFF (toggle)
 - Use Kerberos For Password Authentication: OFF (toggle)
- Sync Settings:
 - Batch Size: 1000
 - Periodic Full Sync: OFF (toggle)
 - Periodic Changed Users Sync: OFF (toggle)
- Cache Settings:
 - Cache Policy: NO_CACHE (dropdown)

Buttons at the bottom: Save, Cancel, Synchronize changed users, Synchronize all users, Remove imported, and Unlink users.

7. To import users to CNCC IAM, Click **Synchronize all users**. If the synchronization is successful, the success message appears. If the synchronization fails, see the troubleshooting section in *Oracle Communication Cloud Native Core Console Installation and Upgrade Guide* and also look at **cncc-iam logs** in debug mode. See [CNC Console Logs](#) for further details.

Figure 4-22 User Federation

The screenshot shows the 'User Federation' settings page. At the top, there is a success message: 'Success! Sync of users finished successfully. Synchronization ignored as it's already in progress.' Below this, there are sections for 'Connection Pooling', 'Kerberos Integration', 'Sync Settings', and 'Cache Settings'. The 'Connection Pooling' section has fields for 'Connection Timeout' and 'Read Timeout', and a 'Connection Pooling Settings' button. The 'Kerberos Integration' section has toggle switches for 'Allow Kerberos authentication' and 'Use Kerberos For Password Authentication', both set to 'OFF'. The 'Sync Settings' section has a 'Batch Size' field set to '1000', and toggle switches for 'Periodic Full Sync' and 'Periodic Changed Users Sync', both set to 'OFF'. The 'Cache Settings' section has a 'Cache Policy' dropdown set to 'NO_CACHE'. At the bottom, there are buttons for 'Save', 'Cancel', 'Synchronize changed users', 'Synchronize all users', 'Remove imported', and 'Unlink users'.

- The user can view the imported users by clicking **Users** under **Manage** in the left pane and click **View all users** in the right pane. The list of users and details appears.

Figure 4-23 User Screen

The screenshot shows the 'User Screen' in the CNC Console IAM. The left pane has a 'Configure' section with 'Realm Settings', 'Clients', 'Client Scopes', 'Roles', 'Identity Providers', 'User Federation', and 'Authentication'. Below this is a 'Manage' section with 'Groups', 'Users', 'Sessions', 'Events', 'Import', and 'Export'. The 'Users' option is selected. The right pane shows a 'Lookup' section with a search bar and a 'View all users' button. Below this is a table of users.

ID	Username	Email	Last Name	First Name	Actions
2290ccb5-b8e9-406f-a8b7-...	admin				Edit Impersonate Delete
b6c14232-1115-454a-a409-...	ben		Alex	Ben Alex	Edit Impersonate Delete
1a5ac7a0-d0dc-44c3-ba9e-...	bob		Hamilton	Bob Hamilton	Edit Impersonate Delete
3a7f9906-2b93-49d8-80cd-7...	joe		Smeth	Joe Smeth	Edit Impersonate Delete

- The user can remove the imported users by clicking the **Remove imported** and set **Import Users** to **OFF** to ensure that the users are not imported to CNCC IAM on your subsequent logins.

Figure 4-24 User Screen

The screenshot displays the 'User Screen' in the CNC Console. It features a dark sidebar on the left and a main content area with various configuration sections. At the top, a green success message states: 'Success! Remove imported users finished successfully.' The sections include:

- Search Scope:** Set to 'One Level'.
- Validate Password Policy:** Toggle set to 'OFF'.
- Trust Email:** Toggle set to 'OFF'.
- Use Truststore SPI:** Set to 'Only for Idps'.
- Connection Pooling:** Toggle set to 'ON'.
- Connection Timeout:** Input field for 'Connection Timeout'.
- Read Timeout:** Input field for 'Read Timeout'.
- Pagination:** Toggle set to 'ON'.
- Kerberos Integration:**
 - Allow Kerberos authentication:** Toggle set to 'OFF'.
 - Use Kerberos For Password Authentication:** Toggle set to 'OFF'.
- Sync Settings:**
 - Batch Size:** Input field set to '1000'.
 - Periodic Full Sync:** Toggle set to 'OFF'.
 - Periodic Changed Users Sync:** Toggle set to 'OFF'.
- Cache Settings:**
 - Cache Policy:** Set to 'NO_CACHE'.

At the bottom, there are buttons for 'Save', 'Cancel', 'Synchronize changed users', 'Synchronize all users', 'Remove imported', and 'Unlink users'. A 'Connection Pooling Settings' link is also visible.

**Note:**

The steps 8 and 9 are optional.

Grouping the LDAP Mapper and Assigning the Roles

When an LDAP Federation provider is created, CNC Console-IAM provides a set of built-in mappers for this provider. User can change the set and create a new mapper or update and delete existing ones.

Group Mapper

The Group Mapper allows you to configure group mappings from LDAP into cncc-iam group mappings. Group mapper can be used to map LDAP groups from a particular branch of an LDAP tree into groups in cncc-iam. It also propagates user-group mappings from LDAP into user-group mappings in cncc-iam.

Perform the following procedure to add the group mapper and assign the roles:

1. Click **Configure** and click **User Federation**. Click **Idap** and select the **Mappers** tab, and click **Create**.

The screenshot shows the 'Configure' page for 'Ldap' in the CNC Console. The 'Mappers' tab is selected. A table lists existing mappers, all of which are 'user-attribute-ldap-mapper' type.

Name	Type
first name	user-attribute-ldap-mapper
last name	user-attribute-ldap-mapper
creation date	user-attribute-ldap-mapper
username	user-attribute-ldap-mapper
modify date	user-attribute-ldap-mapper
email	user-attribute-ldap-mapper

At the bottom right of the table, there is a 'Create' button. The left sidebar shows the navigation menu with 'User Federation' selected.

- The **Add User federation mapper** page appears. Give an appropriate name for the field **Name**. Select 'group-ldap-mapper' as **Mapper Type** drop down menu. Click **Save**.

Configure

- Realm Settings
- Clients
- Client Scopes
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users
- Sessions
- Events
- Import
- Export

Add user federation mapper

Name *

Mapper Type

The following screen appears.

Clients

- Client Scopes
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users
- Sessions
- Events
- Import
- Export

Name *

Mapper Type

LDAP Groups DN

Group Name LDAP Attribute

Group Object Classes

Preserve Group Inheritance

Ignore Missing Groups

Membership LDAP Attribute

Membership Attribute Type

Membership User LDAP Attribute

LDAP Filter

Mode

User Groups Retrieve Strategy

Member-Of LDAP Attribute

Mapped Group Attributes

Note:

When selected, default values will be set by cncc-iam. But you need change some values based on your ldap records.

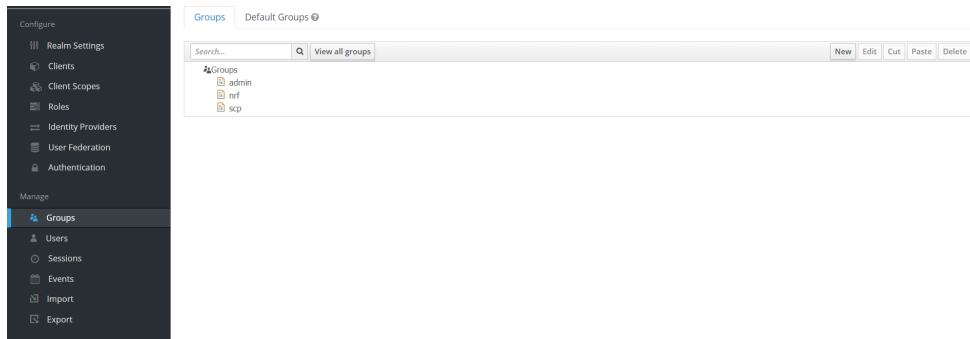
- Click **Save**. New buttons appears next to the **Save** and **Cancel**. They are **Synchronize LDAP Groups to Keycloak** and **Synchronize Keycloak Groups to LDAP**.

4. Click **Synchronize LDAP Groups to Keycloak**. The success message appears with the number of groups imported and so on.

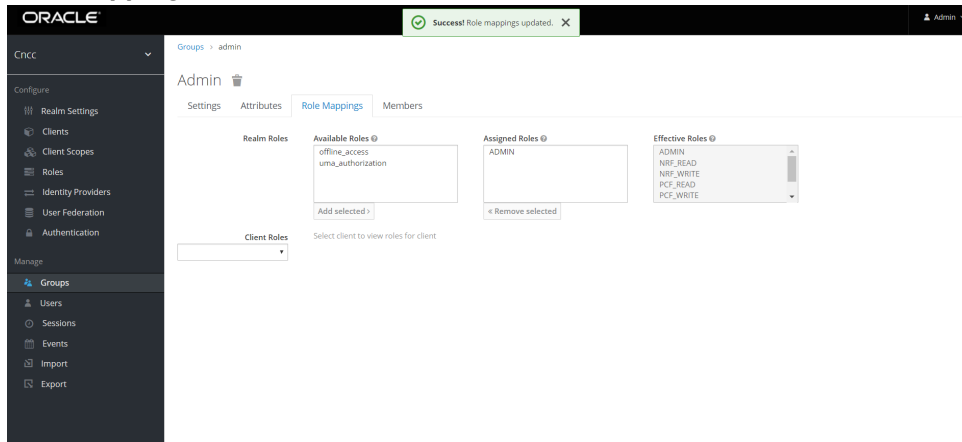
Note:

If this step fails then you might need to check to the trouble shooting section and look at cncc-iam logs in debug mode. See the *Oracle Communication Cloud Native Core Console Troubleshooting Guide* and see [CNC Console Logs](#) for further details.

5. Select the **Groups** in the left pane and click the **View all groups** in the right pane.



6. Click any group and click **Edit**. The following tabs appear: **Settings**, **Attributes**, **Role Mappings**, and **Members**.



- Select **Role Mapping** tab to see a list of roles that are pre-defined in cncc-iam.
- Select one or more roles from **Available Roles** and assign it to the group. For example, If group "admin" is assigned with role "ADMIN", it means that any user which belongs to the admin group will be automatically assigned the admin role which allows him to access all the NF resource of CNC console that it supports.
- Once done you can test authentication and authorization by logging into CNC Console GUI.

Note:

- When the password of user is updated from CNCC-IAM and sent to LDAP, it is always sent in plain-text. This is different from updating the password to built-in CNCC-IAM database, when the hashing and salting is applied to the password before it is sent to DB. In the case of LDAP, the CNCC-IAM relies on the LDAP server to provide hashing and salting of passwords.
- Most of LDAP servers (Microsoft Active Directory, RHDS, FreeIPA) provide this by default. Some others (OpenLDAP, ApacheDS) may store the passwords in plain text by default and user need to explicitly enable password hashing for them.

5

Configuring NF Resources in CNC Console IAM

CNC Console IAM REST API provides a secure option to access NF resources through access token. This chapter describes how to generate and refresh access tokens through CNC Console IAM REST API. You can generate access tokens through the CNC Console IAM REST API by using curl or Postman. Once the access token is generated, CNC Console will be able to access the required NF resources.

Generate Access Tokens

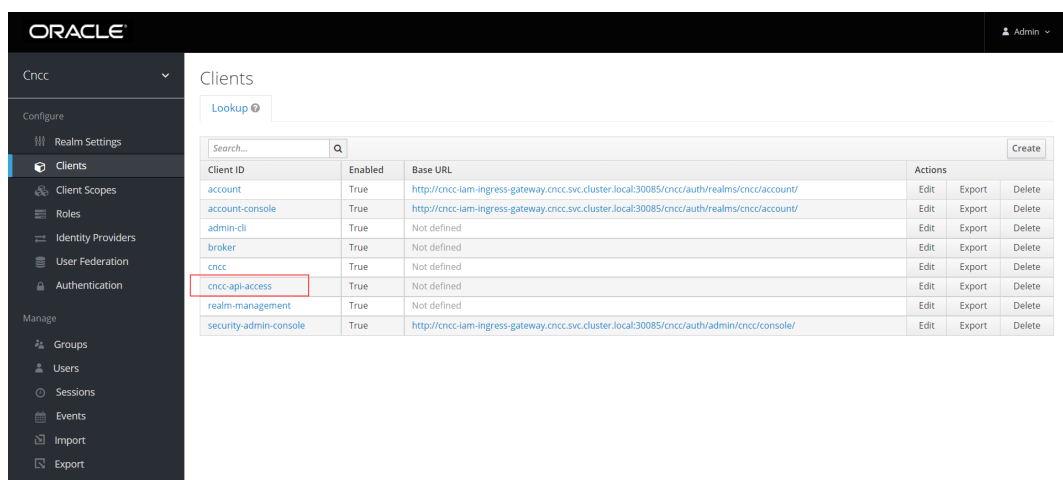
CNCC IAM provides a REST API for generating and refreshing access tokens.

You must use *cncc-api-access* client for accessing NF resources through REST APIs. For security reasons, **Direct Access Grants Enabled** is set to OFF by default.

Perform the following steps to set **Direct Access Grants Enabled** to ON.

1. Log in to CNC IAM with valid credentials.
2. Click **Cncc** realm.
3. On the right pane, click **Clients**. The following screen appears:

Figure 5-1 Clients



Client ID	Enabled	Base URL	Actions
account	True	http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/cncc/auth/realms/cncc/account/	Edit Export Delete
account-console	True	http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/cncc/auth/realms/cncc/account/	Edit Export Delete
admin-cli	True	Not defined	Edit Export Delete
broker	True	Not defined	Edit Export Delete
cncc	True	Not defined	Edit Export Delete
cncc-api-access	True	Not defined	Edit Export Delete
realm-management	True	Not defined	Edit Export Delete
security-admin-console	True	http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/cncc/auth/admin/cncc/console/	Edit Export Delete

4. Click **cncc-api-access**. The following screen appears.

The screenshot shows the Oracle Cloud IAM console interface. On the left is a navigation menu with options like 'Clients', 'Client Scopes', 'Roles', etc. The main panel displays the configuration for a client named 'cncc-api-access'. The 'Settings' tab is active. At the bottom of the settings, the 'Direct Access Grants Enabled' toggle is highlighted with a red rectangular box and is currently set to 'ON'.

5. Set the **Direct Access Grants Enabled** to **ON** and click **Save**.

Perform the following procedure to generate the access tokens:

1. Acquire an access token from CNC Console IAM by sending a POST request to the following URL:

```
http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-port}/cncc/auth/realms/${realm}/protocol/openid-connect/token
```

Example:

```
http://10.75.182.79:8080/cncc/auth/realms/cncc/protocol/openid-connect/token
```

2. The body of the request must be *x-www-form-url* encoded as follows:

```
'client_id': 'your_client_id',
'username': 'your_username',
'password': 'your_password',
'grant_type': 'password'
```

Example:

```
'client_id': 'cncc-api-access',
'username': 'user1',
'password': '*****',
'grant_type': 'password'
```

3. The Curl Command to access the token is as follows:

```
curl --location --request POST 'http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-port}/cncc/auth/realms/cncc/protocol/openid-connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'username=user1' \
--data-urlencode 'password=User123!' \
--data-urlencode 'client_id=cncc-api-access'
```

4. In response, you will get an **access_token** and a **refresh_token**. The response is as follows:

```
{
  "access_token":
    "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJHS1N4WVhoWlExRVhrOVE5R
    TR3STN4WG9LcHI2RW5yOFJCdGlmVndPV0JZIn0.eyJqdGkiOiIwMTQzYzNhZC1kNjE3LTQyNTY
    tYTA2My01NGYxNGI5MDQ5MWMiLCJleHAiOiJlODQ3MDAyMzUsIm5iZiI6MCwiaWF0IjoxNTg0N
    jk5OTM1LCJpc3MiOiJodHRwOi8vMTAuNzUuMjEzLjYwOjMwMDI0L2NuY2MvYXV0aC9yZWZsbXN
    vY25jYyIsImF1ZCI6ImFjY291bnQiLCJzZWVlOiIwMTJlNDI2OS02YzYwLTQzNWItYWExNC0yY
    WI3NmJjOWI1MjMiLCJ0eXAiOiJCZWFyZXIiLCJhenAiOiJhcGktZ2F0ZXdhcSIsImF1ZGhfdG1
    tZSI6MCwic2Vzc2lvdW9zdGF0ZSI6IjZjNDJkOTc4LTE0YWMtNDc5My1hMWUzLTc4OWNmYmRiM
    mI3NCIsImFjciI6IjEiLCJyZWZsbV9hY2Nlc3MiOncicm9sZXMiOlsiUENGX1JFQUUqIiLCJQ0Z
    fVl1JjVEUiLCJQTjBfUkVBRCIsIm9mZmxpbmVfYWNjZXNzIiwiaWF0IjE1JG91dSSVRFIiwiaQURNSU4iL
    CJlbnVfYXV0aG9yaXphdGlvbiIsIk5SRl9SRUFEIiwiaU0NQX1dSSVRFIi19LCJyZXNvdXJjZV9
    hY2Nlc3MiOnciYWNjb3VudCI6eyJyb2xlcjY6WjYtYXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0
    2NvdW50LWxpbnVudCI6eyJyb2xlcjY6WjYtYXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0
    iZWlhaWxfdmVyaWZpZWIiOiJ0eXAiOiJCZWFyZXIiLCJ0eXAiOiJCZWFyZXIiLCJ0eXAiOiJCZWFyZXI
    XJuYXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0
    sImVtYWlsIjoiZ2hyZXlhcy5iQG9yYWNsZS5jb20ifQ.fXYyjmAbSSIFlLr2ZBEX2pfKrE_vr6
    Zbj8ta-
    l_tKlv2gTX1J3ehScg_m30swpWU7UojuFkyc8CfNZL2Z9mcs7zbq_zA7ZTlaWA_AgmeoXWapic
    X2wALT_YDU6Z3H7L9x1C1Ulp8aTBIBHPv2J-
    zgkrFDtk83NeKunKEGLEZpp-9MGDLQ5a8QX6SAUo-
    Fe6hNgFlvP0d7LCyJWvu6UvoeG_Fuxsi4xEVHcbSen8M3eueAt7xN7akhXZ_4PgWnxsWwVQqtT
    zsy60-
    WyUjUiuwtaTvpX0dPVVeeNDvWMy_0q0KvF_nnE3_wQtE8bu_LcCZYwDQJTTloj2PJ8y1Wj0912Q
    ",
    "expires_in": 300,
    "refresh_expires_in": 1800,
    "refresh_token":
      "eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJ3YTFlYjcyZi00MWE1LlRkM
      TETyJmZS01NDZjZGU5NjY2MTUifQ.eyJqdGkiOiJmYjAwZTY2OC0xZTkYLTlMTUtyTVlMS1j
      ZjgxNDk5MjE1NDM1LCJleHAiOiJlODQ3MDc5MzUsIm5iZiI6MCwiaWF0IjoxNTg0Njk5OTM1LC
      Jpc3MiOiJodHRwOi8vMTAuNzUuMjEzLjYwOjMwMDI0L2NuY2MvYXV0aC9yZWZsbXNvY25jYyIs
      ImF1ZCI6Imh0dHA6Ly8xMC43NS4yMTMuNjA6MzAwMjQvY25jYy9hdXR0L3JlYWxtcy9jb2NjIi
      wic3ViIjoiMDEyZTQyNjktNmM2MC00MzViLWFlhMTQzMmFiNzZiYzliNTIzIiwiaWF0IjE1JG91d
      cmVzaCIsImF1ZCI6ImFwaS1nYXRld2F5IiwiaXV0aF90aW1lIjowLCJzZXNzaW9uX3N0YXRlIj
      oiNmM0MmQ5NzgtMTRhYy00NzgzLWExZTMtNzg5Y2ZiZGIyYjc0IiwicmVhbGlfYWNjZXNzIj
      p7InJvbGVzIjpbIlBDRl9SRUFEIiwiaUENGX1dSSVRFIiwiaU0NQX1JFQUUqIiLCJvZmZsaW5lX
      2FjY2VzcyIsIk5SRl9XUklURSI6ImFETU0iLCJ0eXAiOiJCZWFyZXIiLCJ0eXAiOiJCZWFyZXI
      lNDUFE9XUklURSI6ImFETU0iLCJ0eXAiOiJCZWFyZXIiLCJ0eXAiOiJCZWFyZXIiLCJ0eXAiOiJC
      FuYXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0aG9yaXV0
      InNjb3BlIjoiZWlhaWwgcHJvZmlsZS5jb20ifQ.18w3j1gMNgblKSYdvCmJQfg6yIfkdKnMfb5vKPF-
      ZIg",
    "token_type": "bearer",
    "not-before-policy": 0,
    "session_state": "6c42d978-14ac-4793-a1e3-789cfbdb2b74",
    "scope": "email profile"
}
```

 Note:

M-CNCC IAM IP/FQDN which is used to generate access token and M-CNCC IAM IP/FQDN which is specified in *custom-cncc_values.yaml* should be the same.

Refresh Access Tokens

Perform the following procedure to refresh the access tokens:

If the `access_token` has expired, it can be refreshed by sending a POST request to the same URL as above; but the POST method must have the refresh token instead of username and password. The format is as follows:

```
'client_id': 'your_client_id',
'refresh_token': refresh_token_from_previous_request,
'grant type': 'refresh token'
```

Example:

[illegible]

In response, you will receive a new **access_token** and **refresh_token**.

Accessing NF Resources

Perform the following procedure to access NF Resources:

To access NF Resources, you must use the access token in every request to an NF resource by placing it in the Authorization header.

The following headers are to be included while sending API request:

- **Authorization** : The access token should be used in every request to a NF resource by placing it in the *Authorization* header

- `oc-cncc-id` : M-CNCC uses the `oc-cncc-id` header to find the agent or master owning the instance.
- `oc-cncc-instance-id` : A-CNCC Core (or M-CNCC Core) uses the `oc-cncc-instance-id` header to find the NF instance for routing.

Using CNC Console IAM API, the following headers need to be passed in curl/postman request while accessing NF resource:

```
curl --location --request GET 'http://${cncc-mcore-ingress-external-ip}:$
{cncc-mcore-ingress-service-port}/<NF API URI>' \

--header 'oc-cncc-id: oc-cncc-id value' \

--header 'oc-cncc-instance-id: oc-cncc-instance-id value' \

--header 'Authorization: Bearer <token>'
```

Example: Using SCP Canary Release API, the following headers need to be passed in curl/postman request while accessing NF resource:

```
curl --location --request GET 'http://${cncc-mcore-ingress-external-ip}:$
{cncc-mcore-ingress-service-port}/ocscp/scpc-configuration/v1/canaryrelease
' \

--header 'oc-cncc-id: Cluster1' \

--header 'oc-cncc-instance-id: Cluster1-scp-instance1' \

--header 'Authorization: Bearer <token>'
```

6

REST APIs and HTTP Response Codes

The following are the supported REST APIs and HTTP Response Codes:

Generate access token

CNC Console uses `Generate access token` REST API to generate the access token.

Type: POST

URI:

`POST /{realm}/protocol/openid-connect/token`

Table 6-1 Request Body Parameters

Field Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
client_id	string	M	Id that has been given to the client
username	string	M	Login Username
password	string	M	Login Password
grant_type	string	M	Type of Authorization used

Sample URI:

`http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-port}/cncc/auth/realms/master/protocol/openid-connect/token`

Example curl command:

```
curl --location --request POST 'http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-port}/cncc/auth/realms/master/protocol/openid-connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id=admin-cli' \
--data-urlencode 'username=admin' \
--data-urlencode 'password=xxxxxx' \
--data-urlencode 'grant_type=password'
```

Example of the Request Body

The following is the example of the request body:

```
{
  "client_id": "admin-cli",
  "username": "admin",
  "password": "xxxxxxx",
  "grant_type": "password",
}
```

Example of the Response Body

The following is the example of the response body:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIjEiMfag",
  "expires_in": 60,
  "refresh_expires_in": 1800,
  "refresh_token":
"eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIjEiMfag"
  "not-before-policy": 0,
  "session_state": "52dd8d7c-f8d9-4009-9c34-0262bb7d3722",
  "scope": "email profile"
}
```

Table 6-2 Supported Response Codes

Code	Description
200 OK	Get users. Returns a list of users.
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found

Create a new user

CNC Console uses `Create a new user` REST API to create a new user. The user name must be unique.

Type: POST

URI:

POST /{realm}/users

Table 6-3 Request Body Parameters

Field Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
enabled	boolean	M	Set to true to enable the user , an disabled user can not login
username	string	M	Name of the new user

Sample URI:

```
http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-port}/cncc/  
auth/realms/cncc/users
```

Example curl command:

```
{curl --location --request POST 'http://${cncc-iam-ingress-extrenal-ip}:${  
{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users' \  
--header 'Authorization: Bearer <token>' \  
--header 'Content-Type: application/json' \  
--data-raw '{  "enabled": true,  
    "username": "user6"}'
```

Example of the Request Body

The following is the example of the request body:

```
{  "enabled": true,  
    "username": "user6"}
```

Example of the Response Code

The following is the example of the response code:

201 Created

Table 6-4 Supported Response Codes

Code	Description
201 Created	Create a new user.Username must beunique. {Requirespayload}
401 Unauthorised	MissingAuthentication
404 Not Found	Realm not found
409 Conflict	User exists withsame username

Get users

CNC Console uses `Get users` REST API to return a list of users.

Type: GET

URI:

```
GET /{realm}/users
```

Sample URI:

```
http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-  
port}/cncc/auth/realms/cncc/users
```

Example curl command:

```
curl --location --request GET 'http://${cncc-iam-ingress-extrenal-ip}:$  
{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users' \  
--header 'Authorization: Bearer <token>'
```

Example of the Response Body

The following is the example of the response body:

```
[  
  {  
    "id": "48cf183c-d3e4-4917-b3e5-5e01109f534c",  
    "createdTimestamp": 1659952887114,  
    "username": "user",  
    "enabled": true,  
    "emailVerified": false,  
    "access": {  
      "manageGroupMembership": true,  
      "view": true,  
      "mapRoles": true,  
      "impersonate": true,  
      "manage": true  
    }  
  },  
  ....  
]
```

Table 6-5 Supported Response Codes

Code	Description
200 OK	Get users. Returns a list of users, filtered according to query parameters.
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found

Get single user

CNC Console uses `Get single user` REST API to get individual user details.

Type: GET

URI:

```
GET /{realm}/users/{id}
```

Table 6-6 Request Path Parameter

Field Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
id	string	M	Id of the user

Sample URI:

```
http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users/2a3113c0-48de-46d9-a563-6ce95eabb4e4
```

Example curl command:

```
curl --location --request GET 'http://10.75.241.198:30085/cncc/auth/admin/realms/cncc/users/754d6f6b-4ccb-44f1-abf1-00d717885dbe' \
--header 'Authorization: Bearer <token>'
```

Example of the Response Body

The following is the example of the response body:

```
{
  "id": "754d6f6b-4ccb-44f1-abf1-00d717885dbe",
  "createdTimestamp": 1661232491550,
  "username": "ul",
  "enabled": true,
  "totp": false,
  "emailVerified": false,
  "disableableCredentialTypes": [],
  "requiredActions": [],
```

```
"notBefore": 0,  
"access": {  
  "manageGroupMembership": true,  
  "view": true,  
  "mapRoles": true,  
  "impersonate": true,  
  "manage": true  
}  
}
```

Table 6-7 Supported Response Codes

Code	Description
200 OK	Get users. Returns a list of users, filtered according to queryparameters.
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found
404 Not Found	User not found

Delete the User

CNC Console uses Delete the User REST API to delete the user.

Type: DELETE

URI:

```
DELETE /{realm}/users/{id}
```

Table 6-8 Request Path Parameters

Field Name	Data Type	Mandatory(M)/ Optional(O) / Conditional(C)	Description
id	string	M	Id of the user

Sample URI:

```
http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-  
port}/cncc/auth/admin/realms/cncc/users/2a3113c0-48de-46d9-  
a563-6ce95eabbae4
```

Example curl command:

```
curl --location --request DELETE 'http://{cncc-iam-ingress-extrenal-  
ip}:{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users/
```

```
7a2d3608-95b2-4a88-8efb-dad48e7778e2' \--header 'Authorization: Bearer  
<token>'
```

Example of the Response Code

The following is the example of the response code:

```
204 No Content
```

Table 6-9 Supported Response Codes

Code	Description
204 No Content	Delete the user.
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found
404 Not Found	User not found

Set up a New Password for the User

CNC Console uses Setup a New Password for the User REST API to set up a new password for the user.

Type: PUT

URI:

```
PUT /{realm}/users/{id}/reset-password
```

Table 6-10 Request Body Parameters

Field Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
type	string	M	Type
value	Integer	M	Value of new password
temporary	Boolean	O	To validate temporary or not

Table 6-11 Request Path Parameter

Field Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
id	string	M	Id of the user

Sample URI:

```
http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-
port}/cncc/auth/admin/realms/cncc/users/2a3113c0-48de-46d9-
a563-6ce95eabbae4/reset-password
```

Example curl command:

```
curl --location --request PUT 'http://${cncc-iam-ingress-extrenal-ip}:$
{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users/
754d6f6b-4ccb-44f1-abf1-00d717885dbe/reset-password' \
--header 'Authorization: Bearer <token>' \
--header 'Content-Type: application/json' \
--data-raw
'{ "type":"password","value":"Password1#","temporary":false}'
```

Example of the Request Body

The following is the example of the request body:

```
{
  "type":"password",
  "value":"Password1#",
  "temporary":false
}
```

Example of the Response Code

The following is the example of the response code:

```
204 No Content
```

Table 6-12 Supported Response Codes

Code	Description
204 No Content	Set up a newpassword for the CNCC user {Requires payload}.
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found
404 Not Found	User not found
400 Bad Request	Invalid password

Get Realm Level Roles

CNC Console uses Get Realm-level Roles REST API to get realm-level roles.

Type: GET

URI:

GET /{realm}/roles

Sample URI:

http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/roles

Example curl command:

```
curl --location --request GET 'http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/roles' \
--header 'Authorization: Bearer <token>'
```

Example of the Response Body

The following is the example of the response body:

```
[
  {
    "id": "fc5006e0-3927-4034-a01f-af70d779f1f8",
    "name": "ADMIN",
    "description": "Has access to all NF resources and can perform CRUD operations",
    "composite": true,
    "clientRole": false,
    "containerId": "cncc"
  },
  {
    "id": "1acd6c4a-115a-44ae-bf5f-139577f9df0a",
    "name": "POLICY_WRITE",
    "description": "Has access to only POLICY resources and can perform CRUD operation on Managed Objects of POLICY.",
    "composite": true,
    "clientRole": false,
    "containerId": "cncc"
  },
  ...]
```

Table 6-13 Supported Response Codes

Code	Description
200 OK	Generate Get Realm Level Roles

Get Realm-level Role Mappings

CNC Console uses Get Realm-level Role Mappings REST API to get realm-level role mappings for a specific user id .

Type: GET

URI:

GET `/ {realm}/users/{id}/role-mappings/realm`

Table 6-14 Request Path Parameters

Field Name	Data Type	Mandatory(M)/ Optional(O) / Conditional(C)	Description
id	string	M	ID assigned to the user

Sample URI:

`http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users`

Example curl command:

```
curl --location --request GET 'http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/roles' \
--header 'Authorization: Bearer <token>'
```

Example of the Response Body

The following example shows the contents of the response body in JSON format:

```
[
  {
    "id": "fc5006e0-3927-4034-a01f-af70d779f1f8",
    "name": "ADMIN",
    "description": "Has access to all NF resources and can perform
CRUD operations",
    "composite": true,
    "clientRole": false,
    "containerId": "cncc"
  },
]
```

```
{
  "id": "1acd6c4a-115a-44ae-bf5f-139577f9df0a",
  "name": "POLICY_WRITE",
  "description": "Has access to only POLICY resources and can perform
CRUD operation on Managed Objects of POLICY.",
  "composite": true,
  "clientRole": false,
  "containerId": "cncc"
},
...]
```

Table 6-15 Supported Response Codes

Code	Description
200 OK	Get realm-level role mappings for a specific user id.
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found
404 Not Found	User not found

Add Realm-level Role Mappings to the User

CNC Console uses Add Realm-level Role Mappings to the User REST API to add realm-level role mappings to the user.

Type:POST

URI:

POST/{realm}/users/{id}/role mappings/ realm

Table 6-16 Request Body Parameters

Field Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
description	string	M	Description of the role
name	string	M	Name of the role
composite	boolean	M	To check if the role has another realm role mapped to it
clientRole	boolean	M	To check if the role has another client role mapped to it
containerId	string	M	ID of the container where the role is present
id	string	M	ID assigned to the role

Table 6-17 Request Path Parameter

Field Name	Data Type	Mandatory(M)/ Optional(O) / Conditiona l(C)	Description
id	string	M	Id of the user

Sample URI:

```
http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-  
port}/cncc/auth/admin/realms/cncc/users/754d6f6b-4ccb-44f1-  
abf1-00d717885dbe/role-mappings/realm
```

Example curl command:

```
curl --location --request POST 'http://${cncc-iam-ingress-extrenal-  
ip}:${cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users/  
754d6f6b-4ccb-44f1-abf1-00d717885dbe/role-mappings/realm' \  
--header 'Authorization: Bearer <token>' \  
--header 'Content-Type: application/json' \  
--data-raw '[  
  {  
    "id": "c47ba6c5-4cc8-4b59-96bd-2ef7e57121bd",  
    "name": "BSF_READ",  
    "description": "Has access to only BSF resources and can only perform  
    READ Managed Objects of BSF.",  
    "composite": true,  
    "clientRole": false,  
    "containerId": "cncc"  
  }  
'
```

Example of the Request Body

The following is the example of the request body:

```
[  
  {  
    "id": "c47ba6c5-4cc8-4b59-96bd-2ef7e57121bd",  
    "name": "BSF_READ",  
    "description": "Has access to only BSF resources and can only perform  
    READ Managed Objects of BSF.",  
    "composite": true,  
    "clientRole": false,  
    "containerId": "cncc"  
  }  
'
```

Example of the Response Code

The following is the example of the response code:

```
204 No Content
```

Table 6-18 Supported Response Codes

Code	Description
204 No Content	Set up a newpassword for the CNCC user {Requires payload}.
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found
404 Not Found	User not found
404 Not Found	Role not found

Remove All User Sessions Associated with the User

CNC Console uses Remove All User Sessions Associated with the User REST API to remove all user sessions associated with the user.

Type:POST

URI:

```
POST /{realm}/users/{id}/logout
```

Table 6-19 Request Path Parameter

Field Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
id	string	M	Id of the user

Sample URI:

```
http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-port}/cncc/  
auth/admin/realms/cncc/users/48cf183c-d3e4-4917-b3e5-5e01109f534c/logout
```

Example curl command:

```
curl --location --request POST '{cncc-iam-ingress-extrenal-ip}:{cncc-iam-  
ingress-service-port}/cncc/auth/admin/realms/cncc/users/48cf183c-d3e4-4917-  
b3e5-5e01109f534c/logout' \  
--header 'Authorization: Bearer <token>'
```

Example of the Response Code

The following is the example of the response code:

```
204 No Content
```

Table 6-20 Supported Response Codes

Code	Description
204 No Content	Remove all user sessions associated with the user
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found
404 Not Found	User not found

Get the Client details

CNC Console uses `Get the Client details` REST API to get clients belonging to the realm.

Type: GET

URI:

```
GET/{realm}/clients
```

Sample URI:

```
http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-  
port}/cncc/auth/admin/realms/cncc/clients
```

Example curl command:

```
curl --location --request GET '${cncc-iam-ingress-extrenal-ip}:${cncc-  
iam-ingress-service-port}/cncc/auth/admin/realms/cncc/clients' \--  
header 'Authorization: Bearer <token>'
```

Example of the Response Body

The following example shows the contents of the response body in JSON format:

```
[  
  {  
    "id": "b7fa17bd-135f-441b-a5f2-1ea4897e04fc",  
    "clientId": "account",  
    "name": "${client_account}",  
    "rootUrl": "${authBaseUrl}",  
    "baseUrl": "/realms/cncc/account/",  
    "surrogateAuthRequired": false,  
    "enabled": true,  
    "alwaysDisplayInConsole": false,  
    "clientAuthenticatorType": "client-secret",  
    "redirectUris": [  
      ]  
    ]  
  ]  
]
```

```

        "/realms/cncc/account/*"
    ],
    "webOrigins": [],
    "notBefore": 0,
    "bearerOnly": false,
    "consentRequired": false,
    "standardFlowEnabled": true,
    "implicitFlowEnabled": false,
    "directAccessGrantsEnabled": false,
    "serviceAccountsEnabled": false,
    "publicClient": false,
    "frontchannelLogout": false,
    "protocol": "openid-connect",
    "attributes": {},
    "authenticationFlowBindingOverrides": {},
    "fullScopeAllowed": false,
    "nodeReRegistrationTimeout": 0,
    "defaultClientScopes": [
        "web-origins",
        "roles",
        "profile",
        "email"
    ],
    "optionalClientScopes": [
        "address",
        "phone",
        "offline_access",
        "microprofile-jwt"
    ],
    "access": {
        "view": true,
        "configure": true,
        "manage": true
    }
},

```

Update the Client

CNC Console uses `Update the Client` REST API to update the client. This API can be used to update the redirect URI.

Type:PUT

URI:

PUT/{realm}/clients/{id}

Table 6-21 Request Body Parameters

Field Name	Data Type	Mandatory(M)/ Optional(O) / Conditiona I(C)	Description
clientId	string	M	ID given to the client which is referenced in URIs and tokens
rootUri	string	M	Url for redirection

Table 6-22 Request Path Parameter

Field Name	Data Type	Mandatory(M)/ Optional(O) / Conditiona I(C)	Description
id	string	M	ID of the client

Sample URI:

```
http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-  
port}/cncc/auth/admin/realms/cncc/users/clients/9faaa454-  
bbaf-4af0-91dd-2d01aa82776d
```

Example curl command:

```
curl --location --request PUT 'http://${cncc-iam-ingress-extrenal-ip}:$  
{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/clients/  
9faaa454-bbaf-4af0-91dd-2d01aa82776d' \  
--header 'Authorization: Bearer <token>' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
    "clientId": "cncc","rootUrl": "http://10.75.241.74:8080/"  
}'
```

Example of the Request Body

The following is the example of the request body:

```
{  
  "clientId": "cncc",  
  "rootUrl": "http://10.75.241.74:8080/"  
}
```

Example of the Response Code

The following is the example of the response code:

204 No Content

Table 6-23 Supported Response Codes

Code	Description
204 No Content	Update the client.This API can beused to update Redirect URI {Requires payload}.
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found
404 Not Found	Could not find the client

CNC Console Metrics

This section provides the information about CNC Console Metrics.



Note:

Sample CNCC dashboard for Grafana is delivered to the customer through CNCC Custom Templates. Metrics and functions used to achieve KPI are covered in CNCC Custom Templates. Refer to Oracle Help Center site for the information about CNCC Custom Templates.

Two sample dashboards are provided one supporting CNE 1.8 or lower, second one supporting CNE Prometheus HA (OCCNE 1.9.x and higher versions.x and higher versions)

- CNCC Metric Dashboard file : occncc_metric_dashboard_<version>.json
- CNCC Metric Dashboard file supporting CNE Prometheus HA: occncc_metric_dashboard_promha_<version>.json



Note:

Prometheus HA supported CNE (OCOCCNE 1.9.x and higher versions.x and higher versions) tags/labels are renamed as shown in below table

CNE 1.8.0 or previous section	Prometheus HA supported CNE (OCOCCNE 1.9.x and higher versions.x and higher versions)
kubernetes_namespace	namespace
kubernetes_pod_name	pod
container_name	container

Consider updating tags mentioned in following sections of the document as suggested above for Prometheus HA supported CNE

- CNCC IAM Metrics
- CNCC Core Metrics
- CNCC KPIs
- CNCC Alerts

Dimensions Legend for the Metrics

The following table includes the details about the metrics dimensions:

Table 7-1 Dimensions Legend for the Metrics

Dimension	Description	Possible Values
Method	Http method	GET, PUT, POST, DELETE, PATCH
HttpVersion	Http protocol version	HTTP/1.1, HTTP/2.0
Scheme	Http protocol scheme	HTTP, HTTPS, UNKNOWN
Route_Path	Path predicate that matched the current request	NF specific routes
InstanceIdentifier	Identifier for Ingress Gateway	cncc-iam_ingressgateway, cncc-mcore_ingressgateway, cncc-acore_ingressgateway
ResourcePath	Http url	
ResourceType	NF type being accessed	SCP, NRF...
UserId	Id of the user	
UserName	Name of the user	
AuthenticationType	Authentication method used to access CNCC Core GUI	JWT, OAUTH

CNC Console IAM Metrics

This section provides the information about the CNCC IAM Metrics:

CNCC IAM Requests

Table 7-2 CNCC IAM Requests

Metric Details	Total number of requests received by CNCC IAM
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway"}
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

CNCC IAM Response

Table 7-3 CNCC IAM Response

Metric Details	Total number of responses sent by CNCC IAM
----------------	--

Table 7-3 (Cont.) CNCC IAM Response

Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway"}}
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

CNCC IAM Success Responses

Table 7-4 CNCC IAM Success Responses

Metric Details	Total number of success responses (2xx) for CNCC IAM
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"2.*"}
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

CNCC IAM 5xx Responses

Table 7-5 CNCC IAM 5xx Responses

Metric Details	Total number of error responses (5xx) for CNCC IAM
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"5.*"}

Table 7-5 (Cont.) CNCC IAM 5xx Responses

Dimensions	• ErrorOriginator • Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

CNCC IAM 4xx Responses

Table 7-6 CNCC IAM 4xx Responses

Metric Details	Total number of error responses (4xx) for CNCC IAM
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"4.*"}
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

CNCC IAM Error Responses

Table 7-7 CNCC IAM Error Responses

Metric Details	Total number of error responses for CNCC IAM
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"4.* 5.*"}

Table 7-7 (Cont.) CNCC IAM Error Responses

Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

CNCC IAM Access Token Request

Table 7-8 CNCC IAM Access Token Request

Metric Details	Total number of access token requests received for CNCC IAM
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token"} <i>Group by user: (CNE 1.8.x or previous versions)</i> sum by(Status,Method,kubernetes_namespace,ResourceType,UserId,UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token"}) <i>Group by user: (CNE 1.9.0 or later versions)</i> sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token"})
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

CNCC IAM Access Token Granted

Table 7-9 CNCC IAM Access Token Granted

Metric Details	Total number of access token granted for CNCC IAM
Metric Filter	<code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token",Status="200 OK"}</code> <i>Group by user: (CNE 1.8.x or previous versions)</i> <code>sum by(Status,Method,kubernetes_namespace,ResourceType,UserId,UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token",Status="200 OK"})</code> <i>Group by user: (CNE 1.9.0 or later versions)</i> <code>sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token",Status="200 OK"})</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

CNCC IAM Access Token Not Granted

Table 7-10 CNCC IAM Access Token Not Granted

Metric Details	Total number of access token not granted for CNCC IAM
----------------	---

Table 7-10 (Cont.) CNCC IAM Access Token Not Granted

Metric Filter	<pre>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token",Status=~"4.* 5.*"}</pre> <i>Group by user: (CNE 1.8.x or previous versions)</i> <pre>sum by(Status,Method,kubernetes_namespace,ResourceType,UserId, UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token",Status=~"4.* 5.*"})</pre> <i>Group by user: (CNE 1.9.0 or later versions)</i> <pre>sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token",Status=~"4.* 5.*"})</pre>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

CNCC IAM User Login Failure Responses

Table 7-11 CNCC IAM User Login Failure Responses

Metric Details	Total number of user login failure at CNCC IAM
----------------	--

Table 7-11 (Cont.) CNCC IAM User Login Failure Responses

Metric Filter	<code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/login-actions/authenticate",Status="200 OK"}</code> <i>Group by user: (CNE 1.8.x or previous versions)</i> <code>sum by(Status,Method,kubernetes_namespace,ResourceType,UserId,UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/login-actions/authenticate",Status="200 OK"})</code> <i>Group by user: (CNE 1.9.0 or later versions)</i> <code>sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/login-actions/authenticate",Status="200 OK"})</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

CNC Console Core Metrics

This section provides the information about the CNCC Core Metrics:

CNCC Core Requests

Table 7-12 CNCC Core Requests

Metric Details	Total number of requests received by CNCC Core
Metric Filter	<code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway"}</code>

Table 7-12 (Cont.) CNCC Core Requests

Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core Responses

Table 7-13 CNCC Core Responses

Metric Details	Total number of responses for CNCC Core requests
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway"}
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core Success Responses

Table 7-14 CNCC Core Success Responses

Metric Details	Total number of success responses (2xx) for CNCC Core requests
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*"}

Table 7-14 (Cont.) CNCC Core Success Responses

Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core 5xx Responses

Table 7-15 CNCC Core 5xx Responses

Metric Details	Total number of error responses (5xx) for CNCC Core requests
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"5.*"}
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core 4xx Responses

Table 7-16 CNCC Core 4xx Responses

Metric Details	Total number of error responses (4xx) for CNCC Core requests
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.*"}

Table 7-16 (Cont.) CNCC Core 4xx Responses

Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core Error Responses

Table 7-17 CNCC Core Error Responses

Metric Details	Total number of error responses sent for CNCC Core requests
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*"}
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core Access Token Request

Table 7-18 CNCC Core Access Token Request

Metric Details	Total number of access token requests received for CNCC Core
----------------	--

Table 7-18 (Cont.) CNCC Core Access Token Request

Metric Filter	<code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token"}</code> <i>Group by user: (CNE 1.8.x or previous versions)</i> <code>sum by(Status,Method,kubernetes_namespace,ResourceType,UserId,UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token"})</code> <i>Group by user: (CNE 1.9.0 or later versions)</i> <code>sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token"})</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core Access Token Granted Responses

Table 7-19 CNCC Core Access Token Granted Responses

Metric Details	Total number of access token granted for CNCC Core
----------------	--

Table 7-19 (Cont.) CNCC Core Access Token Granted Responses

Metric Filter	<code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status="200 OK"}</code> <i>Group by user:(CNE 1.8.x or previous versions)</i> <code>sum</code> <code>by(Status,Method,kubernetes_namespace,ResourceType,UserId,UserName,kubernetes_pod_name,Scheme)</code> <code>(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status="200 OK"})</code> <i>Group by user: (CNE 1.9.0 or later versions)</i> <code>sum</code> <code>by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme)</code> <code>(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status="200 OK"})</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core Access Token Not Granted Responses

Table 7-20 CNCC Core Access Token Granted Responses

Metric Details	Total number of access token not granted for CNCC Core
----------------	--

Table 7-20 (Cont.) CNCC Core Access Token Granted Responses

Metric Filter	<code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status=~"4.* 5.*"}</code> <i>Group by user: (CNE 1.8.x or previous versions)</i> <code>sum by(Status,Method,kubernetes_namespace,ResourceType,UserId,UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status=~"4.* 5.*"})</code> <i>Group by user: (CNE 1.9.0 or later versions)</i> <code>sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status=~"4.* 5.*"})</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core User Login Failure Responses

Table 7-21 CNCC Core User Login Failure Responses

Metric Details	Total number of user login failure at CNCC Core
----------------	---

Table 7-21 (Cont.) CNCC Core User Login Failure Responses

Metric Filter	<pre>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/login-actions/authenticate",Status="200 OK"}</pre> <i>Group by user: (CNE 1.8.x or previous versions)</i> <pre>sum by(Status,Method,kubernetes_namespace,ResourceType,UserId, UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/login-actions/authenticate",Status="200 OK"})</pre> <i>Group by user: (CNE 1.9.0 or later versions)</i> <pre>sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/login-actions/authenticate",Status="200 OK"})</pre>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core User Authorization Failure Responses

Table 7-22 CNCC Core User Authorization Failure Responses

Metric Details	Total number of authorization failure responses while accessing NF services at CNCC Core
----------------	--

Table 7-22 (Cont.) CNCC Core User Authorization Failure Responses

Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdIdentifier=~".*core_ingressgateway",Status="403 FORBIDDEN", ResourceType!="UNKNOWN"} <i>Group by user: (CNE 1.8.x or previous versions)</i> sum by(Status,Method,kubernetes_namespace,ResourceType,UserId,UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdIdentifier=~".*core_ingressgateway",Status="403 FORBIDDEN", ResourceType!="UNKNOWN"}) <i>Group by user: (CNE 1.9.0 or later versions)</i> sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdIdentifier=~".*core_ingressgateway",Status="403 FORBIDDEN", ResourceType!="UNKNOWN"})
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core BSF Requests

Table 7-23 CNCC Core BSF Requests

Metric Details	Total number of requests received by CNCC Core for BSF
Metric Filter	Single Route_path: oc_ingressgateway_http_requests_total{InstanceIdIdentifier=~".*core_ingressgateway",Route_path=~".*/bsfapi/.*"} Multiple Route_path: oc_ingressgateway_http_requests_total{InstanceIdIdentifier=~".*core_ingressgateway",Route_path=~".*/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*"}

Table 7-23 (Cont.) CNCC Core BSF Requests

Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core BSF Responses

Table 7-24 CNCC Core BSF Responses

Metric Details	Total number of responses sent by CNCC Core for BSF
Metric Filter	Single Route_path: <code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/bsfapi/.*"}</code> Multiple Route_path: <code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*"}</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core DD Requests

Table 7-25 CNCC Core DD Requests

Metric Details	Total number of requests received by CNCC Core for DD
Metric Filter	<code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/ocnadd/.*/ocnaddapi/.*"}</code>

Table 7-25 (Cont.) CNCC Core DD Requests

Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core DD Responses

Table 7-26 CNCC Core DD Responses

Metric Details	Total number of responses sent by CNCC Core for DD
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/ocnadd/.*/ocnaddapi/.*/}
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core NRF Responses

Table 7-27 CNCC Core NRF Responses

Metric Details	Total number of responses sent by CNCC Core for NRF
----------------	---

Table 7-27 (Cont.) CNCC Core NRF Responses

Metric Filter	Single Route_path: <code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/nrf-configuration/v1/*.*/"}</code> Multiple Route_path: <code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/nrf-configuration/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*/nrf-configuration/v1/*.*/"}</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core NRF Requests

Table 7-28 CNCC Core NRF Requests

Metric Details	Total number of requests received by CNCC Core for NRF
Metric Filter	Single Route_path: <code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nrf-configuration/v1/*.*/"}</code> Multiple Route_path: <code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*/"}</code>
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType

Table 7-28 (Cont.) CNCC Core NRF Requests

Metric Type	Counter
-------------	---------

CNCC Core NSSF Requests

Table 7-29 CNCC Core NSSF Requests

Metric Details	Total number of requests received by CNCC Core for NSSF
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nssf-configuration/.*/.*/nssf/.*"}
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core NSSF Responses

Table 7-30 CNCC Core NSSF Responses

Metric Details	Total number of responses sent by CNCC Core for NSSF
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/nssf-configuration/.*/.*/nssf/.*"}
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core POLICY Requests

Table 7-31 CNCC Core POLICY Requests

Metric Details	Total number of requests received by CNCC Core for POLICY
Metric Filter	Single Route_path: <code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/policyapi/.*"}</code> Multiple Route_path: <code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/policyapi/.*/oc-cnpolicy-configuration/.*/pcf/.*"}</code>
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core POLICY Responses

Table 7-32 CNCC Core POLICY Responses

Metric Details	Total number of responses sent by CNCC Core for POLICY
Metric Filter	Single Route_path: <code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/policyapi/.*"}</code> Multiple Route_path: <code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/policyapi/.*/oc-cnpolicy-configuration/.*/pcf/.*"}</code>

Table 7-32 (Cont.) CNCC Core POLICY Responses

Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core SCP Responses

Table 7-33 CNCC Core SCP Responses

Metric Details	Total number of responses sent by CNCC Core for SCP
Metric Filter	Single Route_path: oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/ocscp/.*"}
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core SCP Requests

Table 7-34 CNCC Core SCP Requests

Metric Details	Total number of requests received by CNCC Core for SCP
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/ocscp/.*"}

Table 7-34 (Cont.) CNCC Core SCP Requests

Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core SEPP Requests

Table 7-35 CNCC Core SEPP Requests

Metric Details	Total number of requests received by CNCC Core for SEPP
Metric Filter	Single Route_path: oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/sepp-configuration/.*"} Multiple Route_path: oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/sepp-configuration/.*/sepp/.*"}
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core SEPP Responses

Table 7-36 CNCC Core SEPP Responses

Metric Details	Total number of responses sent by CNCC Core for SEPP
----------------	--

Table 7-36 (Cont.) CNCC Core SEPP Responses

Metric Filter	Single Route_path: oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/sepp-configuration/."} } Multiple Route_path: oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/sepp-configuration/.*/sepp/."} }
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core UDR Requests

Table 7-37 CNCC Core UDR Requests

Metric Details	Total number of requests received by CNCC Core for UDR
Metric Filter	Single Route_path: oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nudr-config/."} } Multiple Route_path: oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*cncc-core_ingressgateway",Route_path=~".*/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/."} }
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

CNCC Core UDR Responses

Table 7-38 CNCC Core UDR Responses

Metric Details	Total number of responses sent by CNCC Core for UDR
Metric Filter	Single Route_path: <pre>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/nudr-dr-prov/.*"}</pre> Multiple Route_path: <pre>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*"}</pre>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

8

CNC Console Alerts

This section provides information about CNC Console Alerts.



Note:

Alert file is packaged with CNCC custom templates. The *occncc_custom_configtemplates_<version>.zip* file can be downloaded from MOS. Unzip the file to get *occncc_alerting_rules_promha_<version>.yaml* file.

- Review the *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.

CNC Console IAM Alerts

This section provides information about CNC Console IAM Alerts.

CncclamTotalIngressTrafficRateAboveMinorThreshold

Table 8-1 CncclamTotalIngressTrafficRateAboveMinorThreshold

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 <i>occncc_alertrules_<version>.yaml</i> is when CNCC IAM Ingress Rate crosses 70 % of 1000 (Maximum ingress request rate)
Severity	Minor

Table 8-1 (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 <i>CncclamTotalIngressTrafficRateAboveMajorThreshold</i> alert is raised. Note: The threshold is configurable in the <i>occncc_alertrules_<version>.yaml</i> file. Steps: 1. Reassess why the CNCC IAM is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

CncclamTotalIngressTrafficRateAboveMajorThreshold

Table 8-2 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 in <i>occncc_alertrules_<version>.yaml</i> is when CNCC IAM Ingress Rate crosses 80 % of 1000 (Maximum ingress request rate)
Severity	Major

Table 8-2 (Cont.) CncclamTotalIngressTrafficRateAboveMajorThreshold

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
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 <i>CncclamTotalIngressTrafficRateAboveCriticalThreshold</i> alert is raised. Note: The threshold is configurable in the <i>occncc_alertrules_<version>.yaml</i> file. Steps: 1. Reassess why the CNCC IAM is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

CncclamTotalIngressTrafficRateAboveCriticalThreshold

Table 8-3 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 <i>occncc_alertrules_<version>.yaml</i> 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)'

Table 8-3 (Cont.) CncclamTotalIngressTrafficRateAboveCriticalThreshold

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 <i>ocncc_alertrules_<version>.yaml</i> file. Steps: 1. Reassess why the CNCC IAM is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

CncclamMemoryUsageCrossedMinorThreshold

Table 8-4 CncclamMemoryUsageCrossedMinorThreshold

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, container_spec_memory_limit_bytes 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 <i>CncclamMemoryUsageCrossedMajorThreshold</i> alert is raised. Note: The threshold is configurable in the <i>ocncc_alertrules_<version>.yaml</i> file. If guidance is required, contact My Oracle Support.

CncclamMemoryUsageCrossedMajorThreshold

Table 8-5 CncclamMemoryUsageCrossedMajorThreshold

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

Table 8-5 (Cont.) CncclamMemoryUsageCrossedMajorThreshold

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, container_spec_memory_limit_bytes 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 Major Threshold or crosses the critical threshold, in which case <i>CncclamMemoryUsageCrossedCriticalThreshold</i> alert shall be raised. Note: The threshold is configurable in the <i>occncc_alertrules_<version>.yaml</i> file. If guidance is required, contact My Oracle Support.

CncclamMemoryUsageCrossedCriticalThreshold

Table 8-6 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, container_spec_memory_limit_bytes 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 <i>occncc_alertrules_<version>.yaml</i> file. If guidance is required, contact My Oracle Support.

CncclamTransactionErrorRateAbove0.1Percent

Table 8-7 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
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 <i>CncclamTransactionErrorRateAbove1Percent</i> 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.

CncclamTransactionErrorRateAbove1Percent

Table 8-8 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 <i>CncclamTransactionErrorRateAbove10Percent</i> 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.

CncclamTransactionErrorRateAbove10Percent

Table 8-9 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
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 <i>CncclamTransactionErrorRateAbove25Percent</i> 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.

CncclamTransactionErrorRateAbove25Percent

Table 8-10 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	The 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 <i>CncclamTransactionErrorRateAbove50Percent</i> 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.

CncclamTransactionErrorRateAbove50Percent

Table 8-11 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
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 <i>occncc_alertrules_<version>.yaml</i> file. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

CncclamIngressGatewayServiceDown

Table 8-12 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.

Table 8-12 (Cont.) CncclamIngressGatewayServiceDown

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.
------------	--

CncclamFailedLogin

Table 8-13 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 <i>alerts.yaml</i> file. If guidance is required, contact My Oracle Support.

AdminUserCreation

Table 8-14 AdminUserCreation

Trigger Condition	If a new admin account is created in the last 5 min
Severity	Warning

Table 8-14 (Cont.) AdminUserCreation

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
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 <i>occncc_alertrules_<version>.yaml</i> file. Login to admin GUI and review the user created. If guidance is required, contact My Oracle Support.

CncclamAccessTokenFailure

Table 8-15 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

Table 8-15 (Cont.) CncclamAccessTokenFailure

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 <i>ocncc_alertrules_<version>.yaml</i> file. If guidance is required, contact My Oracle Support.
------------	---

CNC Console Core Alerts

This section provides the information about CNC Console Core Alerts.

CnccCoreTotalIngressTrafficRateAboveMinorThreshold

Table 8-16 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 <i>cncc_alert_rules.yaml</i> 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 <i>CnccCoreTotalIngressTrafficRateAboveMajorThreshold</i> alert is raised. Note: The threshold is configurable in the <i>ocncc_alertrules_<version>.yaml</i> file. Steps: 1. Reassess why the CNCC Core is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

CnccCoreTotalIngressTrafficRateAboveMajorThreshold

Table 8-17 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
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 rate falls below the Major threshold or when the total traffic rate crosses the Critical threshold, in which case the CnccCoreTotalIngressTrafficRateAboveCriticalThreshold. Note: The threshold is configurable in the <i>alerts.yaml</i> file. Steps: 1. Reassess why the CNCC Core is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

CnccCoreTotalIngressTrafficRateAboveCriticalThreshold

Table 8-18 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

Table 8-18 (Cont.) CnccCoreTotalIngressTrafficRateAboveCriticalThreshold

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

CnccCoreMemoryUsageCrossedMinorThreshold

Table 8-19 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 container_spec_memory_limit_bytes 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.

Table 8-19 (Cont.) CnccCoreMemoryUsageCrossedMinorThreshold

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 <i>occncc_alertrules_<version>.yaml</i> file. If guidance is required, contact My Oracle Support.
------------	---

CnccCoreMemoryUsageCrossedMajorThreshold

Table 8-20 CnccCoreMemoryUsageCrossedMajorThreshold

Trigger Condition	A pod has reached the configured major threshold(80%) of its memory resource limits.
Severity	Major
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 container_spec_memory_limit_bytes 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 <i>occncc_alertrules_<version>.yaml</i> file. If guidance is required, contact My Oracle Support.

CnccCoreMemoryUsageCrossedCriticalThreshold

Table 8-21 CnccCoreMemoryUsageCrossedCriticalThreshold

Trigger Condition	A pod has reached the configured critical threshold (90%) of its memory resource limits
Severity	Critical

Table 8-21 (Cont.) CnccCoreMemoryUsageCrossedCriticalThreshold

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 container_spec_memory_limit_bytes 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 <i>occncc_alertrules_<version>.yaml</i> file. If guidance is required, contact My Oracle Support.

CnccCoreTransactionErrorRateAbove0.1Percent

Table 8-22 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 <i>CnccCoreTransactionErrorRateAbove1Percent</i> is raised. The threshold is configurable in the <i>alerts.yaml</i> file. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

CnccCoreTransactionErrorRateAbove1Percent

Table 8-23 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 <i>CnccCoreTransactionErrorRateAbove10Percent</i> 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.

CnccCoreTransactionErrorRateAbove10Percent

Table 8-24 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 transaction 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 <i>CnccCoreTransactionErrorRateAbove25Percent</i> 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.

CnccCoreTransactionErrorRateAbove25Percent

Table 8-25 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 <i>CnccCoreTransactionErrorRateAbove50Percent</i> 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.

CnccCoreTransactionErrorRateAbove50Percent

Table 8-26 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.

CnccCoreIngressGatewayServiceDown

Table 8-27 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.
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.

CnccCoreFailedLogin

Table 8-28 CnccCoreFailedLogin

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

Table 8-28 (Cont.) CnccCoreFailedLogin

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 <i>alerts.yaml</i> file. If guidance is required, contact My Oracle Support.

CnccCoreUnauthorizedAccess

Table 8-29 CnccCoreUnauthorizedAccess

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

Table 8-29 (Cont.) CnccCoreUnauthorizedAccess

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.
------------	---

CnccCoreAccessTokenFailure

Table 8-30 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 1.8.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' 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'
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 If guidance required, Contact My Oracle Support.

CNCC Alert configuration in Prometheus

CNE 1.8 or previous versions:

This section describes the measurement based Alert rules configuration for CNCC in Prometheus. Use the *ocncc_alertrules_<version>.yaml* file updated in CNCC Alert configuration section.

NAME :- Helm Release of Prometheus

Namespace :- Kubernetes NameSpace in which Prometheus is installed

Configuration

1. Run the following command to take Backup of current config map of prometheus server:

```
kubectl get configmaps _NAME_-server -o yaml -n _Namespace_ > /tmp/tempConfig.yaml
```

where, <Namespace> is the prometheus server namespace used in helm install command.

For Example, assuming chart name is "prometheus-alert", so "_NAME_-server" becomes "prometheus-alert-server", run the following command to find the config map:

```
kubectl get configmaps prometheus-alert-server -o yaml -n prometheus-alert2 > /tmp/tempConfig.yaml
```

2. Run the following command to check and Add CNCC Alert File name inside Prometheus config map:

- a. If alertscncc is present, delete the alertscncc entry from the t_mapConfig.yaml file, by executing the following command:

```
sed -i '/etc/config/alertscncc/d' /tmp/mapConfig.yaml
```

- b. If alertscncc is not present, add the alertscncc entry in the t_mapConfig.yaml file by executing the following command:

```
sed -i '/rule_files:/a\ \ \ - /etc/config/alertscncc' /tmp/mapConfig.yaml
```

3. Run the following command to update Config map with updated file name of CNCC:

```
kubectl replace configmap _NAME_-server -f /tmp/tempConfig.yaml
```

4. Run the following command to add cnccAlertRules.yaml file into prometheus config map under filename of CNCC alert file:

```
kubectl patch configmap _NAME_-server -n _Namespace_ --type merge --patch "$(cat ~/occncc_alertrules_<version>.yaml) "
```

5. Restart prometheus-server pod.
6. Verify the alerts in prometheus GUI.



Note:

Prometheus server takes updated configmap reloaded after sometime automatically (~20 sec)

Example:

```
$ kubectl get configmaps occne-prometheus-server -o yaml -n occne-  
infra > /tmp/tempConfig.yaml  
$ sed -i '/etc\/config\/alertscncc/d' /tmp/tempConfig.yaml  
$ sed -i '/rule_files:/a\ \- /etc/config/alertscncc' /tmp/  
tempConfig.yaml  
$ kubectl replace configmap occne-prometheus-server -f /tmp/  
tempConfig.yaml  
$ kubectl patch configmap occne-prometheus-server -n occne-infra --  
type merge --patch "$(cat ~/occncc_alertrules_<version>.yaml) "
```

**Note:**

The Prometheus server takes an updated configuration map that is automatically reloaded after approximately 60 seconds. Refresh the Prometheus GUI to confirm that the CNCC Alerts have been reloaded.

CNE supporting Prometheus HA (CNE 1.9)

This section describes the measurement based Alert rules configuration for CNCC in Prometheus. Use an updated *occncc_alerting_rules_promha_<version>.yaml* file.

The PrometheusRule resource specified in the alert YAML file can be created or updated by using following command:

```
$ kubectl apply -f ocncnc_alerting_rules_promha_<version>.yaml
```

Validating Alerts

Configure and Validate Alerts in Prometheus Server. Refer to CNCC Alert Configuration section for procedure to configure the alerts.

After configuring the alerts in Prometheus server, a user can verify that by following steps:

1. Open the Prometheus server from your browser using the <IP>:<Port>
2. Navigate to Status and to Rules.
3. Search CNCC. CNCC Alerts list is displayed.

**Note:**

If you are unable to see the alerts, it means the alert file is not loaded in a proper format which the Prometheus server accepts. Modify the file and try again

Disabling Alerts

This section explains how to disable the alerts in CNCC.

1. Edit `occncc_alerting_rules_promha_<version>.yaml` file to remove specific alert.
2. Remove complete content of the specific alert from the `occncc_alerting_rules_promha_<version>.yaml` file.

cncc_alert_rules_<version>.yaml

For example: If you want to remove CnccIamTotalIngressTrafficRateAboveMinorThreshold alert, remove the complete content:

```
## ALERT SAMPLE START##
- alert: CnccIamTotalIngressTrafficRateAboveMinorThreshold
  annotations:
    description: 'CNCC IAM Ingress traffic Rate is above the configured
minor threshold i.e. 700 requests per second (current value is:
{{ $value }})'
    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) '
    expr:
sum(rate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*cncc
-iam_ingressgateway",kubernetes_namespace="cncc"}[2m])) > 0
    labels:
      severity: minor
      oid: "1.3.6.1.4.1.323.5.3.51.1.2.7001"
      namespace: ' {{ $labels.kubernetes_namespace }} '
      podname: ' {{ $labels.kubernetes_pod_name }} '
## ALERT SAMPLE END##
```

3. Perform Alert configuration. See [CNCC Alert Configuration](#) section for details.

Configuring SNMP-Notifier

Configure the IP and port of the SNMP trap receiver in the SNMP Notifier using following procedure:

1. Run the following command to edit the deployment:

```
kubectl edit deploy <snmp_notifier_deployment_name> -n <namespace>
```

Example:

```
$ kubectl edit deploy occne-snmp-notifier -n occne-infra
```

2. Edit the destination as follows:

```
--snmp.destination=<destination_ip>:<destination_port>
```

Example:

```
--snmp.destination=10.75.203.94:162
```

MIB Files for CNCC

There are two MIB files which are used to generate the traps. The user need to update these files along with the Alert file in order to fetch the traps in their environment.

occncc_mib_tc_<version>.mib

This file is considered as CNCC top level mib file, where the Objects and their data types are defined.

occncc_mib_<version>.mib

This file fetches the Objects from the top level mib file and based on the Alert notification, these objects can be selected for display.



Note:

MIB files are packaged along with CNCC Custom Templates. Download the file from MOS. Refer to CNCC Installation guide for more details.

9

CNC Console KPIs

This section provides the information about CNC Console KPIs:

CNC Console IAM KPIs

This section provides the information about CNC Console IAM KPIs:

CNCC IAM Requests

Table 9-1 CNCC IAM Requests

KPI Detail	CNCC IAM Requests
Metric Used for KPI	For CNE 1.8.x or previous versions: - all: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",kubernetes_namespace="\$namespace"}[2m])) For CNE 1.9.0 or later versions: - all:sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",namespace="\$namespace"}[2m]))

CNCC IAM Responses

Table 9-2 CNCC IAM Responses

KPI Detail	CNCC IAM Responses
------------	--------------------

Table 9-2 (Cont.) CNCC IAM Responses

Metric Used for KPI	For CNE 1.8.x or previous versions: - all: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"2.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_error: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"4.* 5.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_4xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"4.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_5xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"5.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) For CNE 1.9.0 or later versions: - all: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"2.*",Route_path=~".*",namespace="\$namespace"}[5m])) - all_error: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"4.* 5.*",Route_path=~".*",namespace="\$namespace"}[5m])) - all_4xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"4.*",Route_path=~".*",namespace="\$namespace"}[5m])) - all_5xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",Status=~"5.*",Route_path=~".*",namespace="\$namespace"}[5m]))
---------------------	--

CNCC IAM Success Rate

Table 9-3 CNCC IAM Success Rate

KPI Detail	CNCC IAM Success Rate (2xx responses divided by total responses)
------------	--

Table 9-3 (Cont.) CNCC IAM Success Rate

Metric Used for KPI	For CNE 1.8.x or previous versions: - all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*iam_ingressgateway"}, \text{Status}=\sim\text{"2.*"}, \text{Route_path}=\sim\text{".*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{Route_path}=\sim\text{".*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ For CNE 1.9.0 or later versions: - all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*iam_ingressgateway"}, \text{Status}=\sim\text{"2.*"}, \text{Route_path}=\sim\text{".*"}, \text{namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{Route_path}=\sim\text{".*"}, \text{namespace}=\text{"\$namespace"}\}[5m]))} * 100$
---------------------	--

CNCC IAM Error Rate

Table 9-4 CNCC IAM Error Rate

KPI Detail	CNCC IAM Error Rate (4xx or 5xx responses divided by total responses) for all as well as specific NFs
Metric Used for KPI	For CNE 1.8.0 or previous versions: - all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*iam_ingressgateway"}, \text{Status}=\sim\text{"4.* 5.*"}, \text{Route_path}=\sim\text{".*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{Route_path}=\sim\text{".*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ For CNE 1.9.0 or later versions: - all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*iam_ingressgateway"}, \text{Status}=\sim\text{"4.* 5.*"}, \text{Route_path}=\sim\text{".*"}, \text{namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{Route_path}=\sim\text{".*"}, \text{namespace}=\text{"\$namespace"}\}[5m]))} * 100$

CNC Console Core KPIs

This section provides the information about CNC Console Core KPIs:

CNCC Core Requests

Table 9-5 CNCC Core Requests

KPI Detail	CNCC Core Requests for NFs and specific NFs
Metric Used for KPI	For CNE 1.8.x or previous versions: - all: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",kubernetes_namespace="\$namespace"}[2m])) For CNE 1.9.0 or later versions: - all: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",namespace="\$namespace"}[2m])) - scp: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/ocscp/.*" } [2m])) - nrf: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nrf-configuration/v1.**/nrf-state-data/*.**/ocnrf-swagger/*.**/nrf-status-data/*.**/nrf/nf-common-component/*.*" } [2m])) - udr: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nudr-dr-prov/*.**/nudr-dr-mgm/*.**/nudr-group-id-map-prov/*.**/slf-group-prov/*.**/nudr-config/*.**/udr/nf-common-component/*.**/n5g-eir-prov/*.*" } [2m])) - policy: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/policyapi/*.**/oc-cnpolicy-configuration/*.**/pcf/*.*" } [2m])) - bsf: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/bsfapi/*.**/oc-bsf-configuration/*.**/bsf/*.*" } [2m])) - sepp: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/sepp-configuration/*.**/sepp/*.*" } [2m])) - nssf: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nssf-configuration/*.**/nssf/*.*" } [2m])) - dd: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/ocnadd/*.**/ocnaddapi/*.*" } [2m]))

CNCC Core Responses

Table 9-6 CNCC Core Responses

KPI Detail	CNCC Core Responses for all as well as specific NFs
------------	---

Table 9-6 (Cont.) CNCC Core Responses

Metric Used for KPI	For CNE 1.8.x or previous versions: - all_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_error:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_4xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_5xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"5.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - scp_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~"/ocscp/*",kubernetes_namespace="\$namespace"}[5m])) - nrf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~"/nrf-configuration/v1.*/*nrf-state-data/*/*ocnrf-swagger/*/*nrf-status-data/*/*nrf/nf-common-component/*",kubernetes_namespace="\$namespace"}[5m])) - udr_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~"/nudr-dr-prov/*/*nudr-dr-mgm/*/*nudr-group-id-map-prov/*/*slf-group-prov/*/*nudr-config/*/*udr/nf-common-component/*/*n5g-eir-prov/*",kubernetes_namespace="\$namespace"}[5m])) - policy_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~"/policyapi/*/*oc-cnpolicy-configuration/*/*pcf/*",kubernetes_namespace="\$namespace"}[5m])) - bsf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~"/bsfapi/*/*oc-bsf-configuration/*/*bsf/*",kubernetes_namespace="\$namespace"}[5m])) - sepp_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~"/sepp-configuration/*/*sepp/*",kubernetes_namespace="\$namespace"}[5m])) - nssf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~"/nssf-configuration/*/*nssf/*",kubernetes_namespace="\$namespace"}[5m])) - dd_success:
---------------------	--

Table 9-6 (Cont.) CNCC Core Responses

	<pre>sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/ocnadd/.*/ocnaddapi/.*",kubernetes_namespace="\$namespace"}[5m]))</pre> For CNE 1.9.0 and later versions: - all_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*",namespace="\$namespace"}[5m])) - all_error:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*",Route_path=~".*",namespace="\$namespace"}[5m])) - all_4xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.*",Route_path=~".*",namespace="\$namespace"}[5m])) - all_5xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"5.*",Route_path=~".*",namespace="\$namespace"}[5m])) - scp_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/ocscp/.*",namespace="\$namespace"}[5m])) - nrf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/nrf-configuration/v1.* /nrf-state-data.* /ocnrf-swagger/* /nrf-status-data/* /nrf/nf-common-component/*",namespace="\$namespace"}[5m])) - udr_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/nudr-dr-prov.* /nudr-dr-mgm/* /nudr-group-id-map-prov/* /slf-group-prov/* /nudr-config/* /udr/nf-common-component/* /n5g-eir-prov/*",namespace="\$namespace"}[5m])) - policy_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/policyapi/* /oc-cnpolicy-configuration/* /pcf/*",namespace="\$namespace"}[5m])) - bsf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/bsfapi/* /oc-bsf-configuration/* /bsf/*",namespace="\$namespace"}[5m])) - sepp_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/sepp-configuration/* /sepp/*",namespace="\$namespace"}[5m])) - nssf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path
--	--

Table 9-6 (Cont.) CNCC Core Responses

	<pre>=~".*/nnssf-configuration/.*/ nnssf/.*/",namespace="\$namespace"}[5m])) • dd_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceI dentifier=~".*core_ingressgateway",Status=~"2.*",Route_path =~".*/ocnadd/.*/.*ocnaddapi/.*/",namespace="\$namespace"} [5m]))</pre>
--	--

CNCC Core Success Rate

Table 9-7 CNCC Core Success Rate

KPI Detail	CNCC Core Success Rate (2xx responses divided by total responses) for all as well as specific NFs
------------	---

Table 9-7 (Cont.) CNCC Core Success Rate

Metric Used for KPI	For CNE 1.8.x or previous versions:
	- all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstancelIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{"2."}^*, \text{Route_path}=\sim\text{.}^*, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{Route_path}=\sim\text{.}^*, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - scp: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstancelIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{"2."}^*, \text{Route_path}=\sim\text{.}^*/\text{ocscp}/\text{.}^*, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstancelIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Route_path}=\sim\text{.}^*/\text{ocscp}/\text{.}^*, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - nrf: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstancelIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{"2."}^*, \text{Route_path}=\sim\text{.}^*/\text{nrf-configuration/v1}/\text{.}^*/\text{nrf-state-data}/\text{.}^*/\text{ocnrf-swagger}/\text{.}^*/\text{nrf-status-data}/\text{.}^*/\text{nrf/nf-common-component}/\text{.}^*, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstancelIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Route_path}=\sim\text{.}^*/\text{nrf-configuration/v1}/\text{.}^*/\text{nrf-state-data}/\text{.}^*/\text{ocnrf-swagger}/\text{.}^*/\text{nrf-status-data}/\text{.}^*/\text{nrf/nf-common-component}/\text{.}^*, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - udr: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstancelIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{"2."}^*, \text{Route_path}=\sim\text{.}^*/\text{nudr-dr-prov}/\text{.}^*/\text{nudr-dr-mgm}/\text{.}^*/\text{nudr-group-id-map-prov}/\text{.}^*/\text{slf-group-prov}/\text{.}^*/\text{nudr-config}/\text{.}^*/\text{udr/nf-common-component}/\text{.}^*/\text{n5g-eir-prov}/\text{.}^*, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstancelIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Route_path}=\sim\text{.}^*/\text{nudr-dr-prov}/\text{.}^*/\text{nudr-dr-mgm}/\text{.}^*/\text{nudr-group-id-map-prov}/\text{.}^*/\text{slf-group-prov}/\text{.}^*/\text{nudr-config}/\text{.}^*/\text{udr/nf-common-component}/\text{.}^*/\text{n5g-eir-prov}/\text{.}^*, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - policy: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstancelIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{"2."}^*, \text{Route_path}=\sim\text{.}^*/\text{policyapi}/\text{.}^*/\text{oc-cnpolicy-configuration}/\text{.}^*/\text{pcf}/\text{.}^*, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstancelIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Route_path}=\sim\text{.}^*/\text{policyapi}/\text{.}^*/\text{oc-cnpolicy-configuration}/\text{.}^*/\text{pcf}/\text{.}^*, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$

Table 9-7 (Cont.) CNCC Core Success Rate

	<pre>configuration/.*/pcf/.*/kubernetes_namespace="\$namespace"}[5m]))*100</pre> - bsf: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstancelIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*/kubernetes_namespace="\$namespace"}[5m]))/sum(rate(oc_ingressgateway_http_responses_total{InstancelIdentifier=~".*core_ingressgateway",Route_path=~".*/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*/kubernetes_namespace="\$namespace"}[5m]))*100</pre> - sepp: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstancelIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/sepp-configuration/.*/sepp/.*/kubernetes_namespace="\$namespace"}[5m]))/sum(rate(oc_ingressgateway_http_responses_total{InstancelIdentifier=~".*core_ingressgateway",Route_path=~".*/sepp-configuration/.*/sepp/.*/kubernetes_namespace="\$namespace"}[5m]))*100</pre> - nssf: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstancelIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/nssf-configuration/.*/nssf/.*/kubernetes_namespace="\$namespace"}[5m]))/sum(rate(oc_ingressgateway_http_responses_total{InstancelIdentifier=~".*core_ingressgateway",Route_path=~".*/nssf-configuration/.*/nssf/.*/kubernetes_namespace="\$namespace"}[5m]))*100</pre> - dd: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstancelIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/ocnadd/.*/ocnaddapi/.*/kubernetes_namespace="\$namespace"}[5m]))/sum(rate(oc_ingressgateway_http_responses_total{InstancelIdentifier=~".*core_ingressgateway",Route_path=~".*/ocnadd/.*/ocnaddapi/.*/kubernetes_namespace="\$namespace"}[5m]))*100</pre> For CNE 1.9.0 and later versions: - all: <pre>sum(increase(oc_ingressgateway_http_responses_total{InstancelIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*",namespace="\$namespace"}[5m]))/sum(increase(oc_ingressgateway_http_responses_total{Route_path=~".*",namespace="\$namespace"}[5m]))*100</pre> - scp: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstancelIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/ocscp/.*/kubernetes_namespace="\$namespace"}[5m]))</pre>
--	--

Table 9-7 (Cont.) CNCC Core Success Rate

	<pre>[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Route_path=~". */ocscpl.*",namespace="\$namespace"}[5m]))*100</pre> <pre>nrf: sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Status=~"2.*", Route_path=~".*/nrf-configuration/v1/*.**/nrf-state- data/*.**/ocnrf-swagger/*.**/nrf-status-data/*.**/nrf/nf- common-component/*.*",namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Route_path=~". */nrf-configuration/v1/*.**/nrf-state-data/*.**/ocnrf- swagger/*.**/nrf-status-data/*.**/nrf/nf-common- component/*.*",namespace="\$namespace"}[5m]))*100</pre> <pre>udr: sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Status=~"2.*", Route_path=~".*/nudr-dr-prov/*.**/nudr-dr-mgm/*.**/ nudr-group-id-map-prov/*.**/slf-group-prov/*.**/nudr- config/*.**/udr/nf-common-component/*.**/n5g-eir- prov/*.*",namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Route_path=~". */nudr-dr-prov/*.**/nudr-dr-mgm/*.**/nudr-group-id- map-prov/*.**/slf-group-prov/*.**/nudr-config/*.**/udr/nf- common-component/*.**/n5g-eir- prov/*.*",namespace="\$namespace"}[5m]))*100</pre> <pre>policy: sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Status=~"2.*", Route_path=~".*/policyapi/*.**/oc-cnpolicy- configuration/*.**/pcf/*.*",namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Route_path=~". */policyapi/*.**/oc-cnpolicy- configuration/*.**/pcf/*.*",namespace="\$namespace"} [5m]))*100</pre> <pre>bsf: sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Status=~"2.*", Route_path=~".*/bsfapi/*.**/oc-bsf- configuration/*.**/bsf/*.*",namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Route_path=~". */bsfapi/*.**/oc-bsf- configuration/*.**/bsf/*.*",namespace="\$namespace"} [5m]))*100</pre> <pre>sepp: sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Status=~"2.*", Route_path=~".*/sepp-configuration/*.**/ sepp/*.*",namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Inst</pre>
--	--

Table 9-7 (Cont.) CNCC Core Success Rate

	<pre> ancelIdentifier=~".*core_ingressgateway",Route_path=~".*/sepp-configuration/.*/.*"/ sepp/.*/",namespace="\$namespace"){5m}))*100 </pre> <pre> nssf: sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Status=~"2.*", Route_path=~".*/nssf-configuration/.*/.*"/ nssf/.*/",namespace="\$namespace"){5m}))/ sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Route_path=~".*/nssf-configuration/.*/.*"/ nssf/.*/",namespace="\$namespace"){5m}))*100 </pre> <pre> dd: sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Status=~"2.*", Route_path=~".*/ocnadd/.*/.*"/ ocnaddapi/.*/",namespace="\$namespace"){5m}))/ sum(rate(oc_ingressgateway_http_responses_total{Inst ancelIdentifier=~".*core_ingressgateway",Route_path=~".*/ocnadd/.*/.*"/ ocnaddapi/.*/",namespace="\$namespace"){5m}))*100 </pre>
--	--

CNCC Core Error Rate

Table 9-8 CNCC Core Error Rate

KPI Detail	CNCC Core Error Rate (4xx or 5xx responses divided by total responses) for all as well as specific NFs
------------	--

Table 9-8 (Cont.) CNCC Core Error Rate

Metric Used for KPI	For CNE 1.8.x or previous versions:
	- all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{4.}^* 5.^*\text{,Route_path}=\sim\text{.}^*\text{,kubernetes_namespace}=\text{"\$namespace"}\}[5\text{m}]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Route_path}=\sim\text{.}^*\text{,kubernetes_namespace}=\text{"\$namespace"}\}[5\text{m}]))} * 100$ - scp: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{4.}^* 5.^*\text{,Route_path}=\sim\text{.}^*/\text{ocscp}/.^*\text{,kubernetes_namespace}=\text{"\$namespace"}\}[5\text{m}]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Route_path}=\sim\text{.}^*/\text{ocscp}/.^*\text{,kubernetes_namespace}=\text{"\$namespace"}\}[5\text{m}]))} * 100$ - nrf: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{4.}^* 5.^*\text{,Route_path}=\sim\text{.}^*/\text{nrf-configuration/v1}/.^*/\text{nrf-state-data}/.^*/\text{ocnrf-swagger}/.^*/\text{nrf-status-data}/.^*/\text{nrf/nf-common-component}/.^*\text{,kubernetes_namespace}=\text{"\$namespace"}\}[5\text{m}]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Route_path}=\sim\text{.}^*/\text{nrf-configuration/v1}/.^*/\text{nrf-state-data}/.^*/\text{ocnrf-swagger}/.^*/\text{nrf-status-data}/.^*/\text{nrf/nf-common-component}/.^*\text{,kubernetes_namespace}=\text{"\$namespace"}\}[5\text{m}]))} * 100$ - udr: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{4.}^* 5.^*\text{,Route_path}=\sim\text{.}^*/\text{nudr-dr-prov}/.^*/\text{nudr-dr-mgm}/.^*/\text{nudr-group-id-map-prov}/.^*/\text{slf-group-prov}/.^*/\text{nudr-config}/.^*/\text{udr/nf-common-component}/.^*/\text{n5g-eir-prov}/.^*\text{,kubernetes_namespace}=\text{"\$namespace"}\}[5\text{m}]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Route_path}=\sim\text{.}^*/\text{nudr-dr-prov}/.^*/\text{nudr-dr-mgm}/.^*/\text{nudr-group-id-map-prov}/.^*/\text{slf-group-prov}/.^*/\text{nudr-config}/.^*/\text{udr/nf-common-component}/.^*/\text{n5g-eir-prov}/.^*\text{,kubernetes_namespace}=\text{"\$namespace"}\}[5\text{m}]))} * 100$ - policy: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{4.}^* 5.^*\text{,Route_path}=\sim\text{.}^*/\text{policyapi}/.^*/\text{oc-cnpolicy-configuration}/.^*/\text{pcf}/.^*\text{,kubernetes_namespace}=\text{"\$namespace"}\}[5\text{m}]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{.}^*\text{core_ingressgateway}, \text{Status}=\sim\text{4.}^* 5.^*\text{,Route_path}=\sim\text{.}^*/\text{policyapi}/.^*/\text{oc-cnpolicy-configuration}/.^*/\text{pcf}/.^*\text{,kubernetes_namespace}=\text{"\$namespace"}\}[5\text{m}]))} * 100$

Table 9-8 (Cont.) CNCC Core Error Rate

	<pre> InstanceIdentifier=~".*core_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/policyapi/.*/oc-cnpolicy-configuration/.*/pcf/.*",kubernetes_namespace="\$namespace"}[5m]))*100 </pre> - bsf: <pre> sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*",Route_path=~".*/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*",kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*",kubernetes_namespace="\$namespace"}[5m]))*100 </pre> - sepp: <pre> sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*",Route_path=~".*/sepp-configuration/.*/sepp/.*",kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/sepp-configuration/.*/sepp/.*",kubernetes_namespace="\$namespace"}[5m]))*100 </pre> - nssf: <pre> sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*",Route_path=~".*/nssf-configuration/.*",kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nssf-configuration/.*",kubernetes_namespace="\$namespace"}[5m]))*100 </pre> - dd: <pre> sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*",Route_path=~".*/ocnadd/.*/ocnaddapi/.*",kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/ocnadd/.*/ocnaddapi/.*",kubernetes_namespace="\$namespace"}[5m]))*100 </pre> For CNE 1.9.0 or later versions: - all: <pre> sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*",Route_path=~".*",namespace="\$namespace"} </pre>
--	--

Table 9-8 (Cont.) CNCC Core Error Rate

	<pre>[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{ InstanceIdentifier=~".*core_ingressgateway",Route_path =~".*",namespace="\$namespace"}[5m]))*100</pre> <pre>scp: sum(increase(oc_ingressgateway_http_responses_total{ InstanceIdentifier=~".*core_ingressgateway",Status=~"4. * 5.*",Route_path=~".*/soothsayer/v1/.*/.*/ ocscp/.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{ InstanceIdentifier=~".*core_ingressgateway",Route_path =~".*/ocscp/.*",namespace="\$namespace"}[5m]))*100</pre> <pre>nrf: sum(increase(oc_ingressgateway_http_responses_total{ InstanceIdentifier=~".*core_ingressgateway",Status=~"4. * 5.*",Route_path=~".*/nrf-configuration/v1/.*/.*/nrf-state- data/.*/ocnrf-swagger/.*/nrf-status-data/.*/.*/nrf/nf- common-component/.*",namespace="\$namespace"} [5m]))/ sum(increase(oc_ingressgateway_http_responses_total{ InstanceIdentifier=~".*core_ingressgateway",Route_path =~".*/nrf-configuration/v1/.*/.*/nrf-state-data/.*/.*/ocnrf- swagger/.*/.*/nrf-status-data/.*/.*/nrf/nf-common- component/.*",namespace="\$namespace"}[5m]))*100</pre> <pre>udr: sum(increase(oc_ingressgateway_http_responses_total{ InstanceIdentifier=~".*core_ingressgateway",Status=~"4. * 5.*",Route_path=~".*/nudr-dr-prov/.*/.*/nudr-dr- mgm/.*/.*/nudr-group-id-map-prov/.*/.*/slf-group- prov/.*/.*/nudr-config/.*/.*/udr/nf-common- component/.*/.*/n5g-eir- prov/.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{ InstanceIdentifier=~".*core_ingressgateway",Route_path =~".*/nudr-dr-prov/.*/.*/nudr-dr-mgm/.*/.*/nudr-group-id- map-prov/.*/.*/slf-group-prov/.*/.*/nudr-config/.*/.*/udr/nf- common-component/.*/.*/n5g-eir- prov/.*",namespace="\$namespace"}[5m]))*100</pre> <pre>policy: sum(increase(oc_ingressgateway_http_responses_total{ InstanceIdentifier=~".*core_ingressgateway",Status=~"4. * 5.*",Route_path=~".*/policyapi/.*/.*/oc-cnpolicy- configuration/.*/.*/pcf/.*",namespace="\$namespace"} [5m]))/ sum(increase(oc_ingressgateway_http_responses_total{ InstanceIdentifier=~".*core_ingressgateway",InstanceIde ntifier=~".*core_ingressgateway",Route_path=~".*/ policyapi/.*/.*/oc-cnpolicy- configuration/.*/.*/pcf/.*",namespace="\$namespace"} [5m]))*100</pre> <pre>bsf: sum(increase(oc_ingressgateway_http_responses_total{ InstanceIdentifier=~".*core_ingressgateway",Status=~"4. * 5.*",Route_path=~".*/bsfapi/.*/.*/oc-bsf- configuration/.*/.*/bsf/.*",namespace="\$namespace"} [5m]))/ sum(increase(oc_ingressgateway_http_responses_total{</pre>
--	--

Table 9-8 (Cont.) CNCC Core Error Rate

	<pre> Instanceldentifier=~".*core_ingressgateway",Instancelde ntifier=~".*core_ingressgateway",Route_path=~".*/ bsfapi.* */oc-bsf- configuration/.*/.* */bsf/.*",namespace="\$namespace"} [5m]))*100 </pre> <pre> sepp: sum(increase(oc_ingressgateway_http_responses_total{ Instanceldentifier=~".*core_ingressgateway",Status=~"4. * 5.*",Route_path=~".*/sepp-configuration/.*/.* */ sepp/.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{ Instanceldentifier=~".*core_ingressgateway",Instancelde ntifier=~".*core_ingressgateway",Route_path=~".*/sepp- configuration/.*/.* */sepp/.*",namespace="\$namespace"} [5m]))*100 </pre> <pre> nssf: sum(increase(oc_ingressgateway_http_responses_total{ Instanceldentifier=~".*core_ingressgateway",Status=~"4. * 5.*",Route_path=~".*/nssf-configuration/.*/.* */ nssf/.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{ Instanceldentifier=~".*core_ingressgateway",Instancelde ntifier=~".*core_ingressgateway",Route_path=~".*/nssf- configuration/.*/.* */nssf/.*",namespace="\$namespace"} [5m]))*100 </pre> <pre> dd: sum(increase(oc_ingressgateway_http_responses_total{ Instanceldentifier=~".*core_ingressgateway",Status=~"4. * 5.*",Route_path=~".*/ocnadd/.*/.* */ ocnaddapi/.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{ Instanceldentifier=~".*core_ingressgateway",Instancelde ntifier=~".*core_ingressgateway",Route_path=~".*/ ocnadd/.*/.* */ocnaddapi/.*",namespace="\$namespace"} [5m]))*100 </pre>
--	--

CNC Console Common KPIs

This section provides the information about CNC Console common KPIs:

CPU Usage

Table 9-9 CPU Usage

KPI Detail	CPU usage by all as well as individual microservices in CNCC
------------	--

Table 9-9 (Cont.) CPU Usage

KPI Detail	Memory usage by all as well as individual microservices in CNCC
Metric Used for KPI	- all: sum(rate(container_cpu_usage_seconds_total{pod=~".*", namespace="\$namespace"}[2m])) - cncc-core-cmservice: sum(rate(container_cpu_usage_seconds_total{pod=~".*", cncc-core-cmservice.*", namespace="\$namespace"}[2m])) - cncc-core-ingress: sum(rate(container_cpu_usage_seconds_total{pod=~".*", core-ingress.*", namespace="\$namespace"}[2m])) - cncc-iam-ingress: sum(rate(container_cpu_usage_seconds_total{pod=~".*", iam-ingress.*", namespace="\$namespace"}[2m])) - cncc-iam-kc: sum(rate(container_cpu_usage_seconds_total{pod=~".*", iam-kc.*", namespace="\$namespace"}[2m]))

Memory Usage

Table 9-10 Memory Usage

KPI Detail	Memory usage by all as well as individual microservices in CNCC
Metric Used for KPI	- all: sum (container_memory_usage_bytes{pod=~".*", namespace="\$namespace"}) - cncc-core-cmservice: sum (container_memory_usage_bytes{pod=~".*", core-cmservice.*", namespace="\$namespace"}) - cncc-core-ingress: sum (container_memory_usage_bytes{pod=~".*", core-ingress.*", namespace="\$namespace"}) - cncc-iam-ingress: sum (container_memory_usage_bytes{pod=~".*", iam-ingress.*", namespace="\$namespace"}) - cncc-iam-kc: sum (container_memory_usage_bytes{pod=~".*", iam-kc.*", namespace="\$namespace"})

10

CNC Console Logs

This section describes about the cncc logs. It contains the following topics:

[Log Formats and Details](#)

[Types of Logs](#)

[Configuring the Logs](#)

[Examples of Logs](#)

[Accessing the Logs](#)

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 10-1 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"

Table 10-1 (Cont.) Log Details

Name	Description	Example
loggerName	Name of the class that generated the log	"loggerName": "ocpm.cne.gateway.cncc.Gateway Application"
logTimestamp	Time represented in human readable format and in UTC. Format is <i>date:yyyy-MM-dd'T'HH:mm:ss.SSSZ</i> 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 10-2 CNCC Message Format

Name	Description	Example	Possible Values
logType	Indicates whether it is Security Log or Audit Log	logType=AUDIT	AUDIT SECURITY

Table 10-2 (Cont.) CNCC Message Format

Name	Description	Example	Possible Values
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 10-2 (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

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-asure-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
```

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.

Examples of Logs

This section lists the examples of audit and security logs.

Examples of Audit Logs

Examples of Security Logs

Examples of Audit Logs

CNCC Core

Only message part of the JSON log is shown in the example.

- User successfully logging into CNCC Core

```
logType=AUDIT, type=LOGIN, resourceType=CNCC,
userId=186f6f2a-ba6a-4812-8a18-b906a5f9e3f6, username=shreb,
operationType=GET, remoteAddress=/192.168.219.64:53587,
localAddress=cncc-core-ingress-gateway.cncc.svc.cluster.local/
<unresolved>:30075,
resourcePath=/login/oauth2/code/cncc-iam,
authenticationType=OAUTH
```

- User accessing SCP resource having SCP_READ role

```
logType=AUDIT, type=ACCESS_RESOURCE, resourceType=SCP,
userId=186f6f2a-ba6a-4812-8a18-b906a5f9e3f6, username=shreb,
operationType=GET, remoteAddress=/192.168.219.64:53587,
localAddress=cncc-core-ingress-gateway.cncc.svc.cluster.local/
<unresolved>:30075,
resourcePath=/soothsayer/v1/canaryrelease/,
authenticationType=OAUTH
```

- User updating(PATCH) SCP resource having SCP_WRITE role

```
logType=AUDIT, type=ACCESS_RESOURCE, resourceType=SCP,
userId=186f6f2a-ba6a-4812-8a18-b906a5f9e3f6, username=shreb,
operationType=PATCH, remoteAddress=/192.168.219.64:53587,
localAddress=cncc-core-ingress-gateway.cncc.svc.cluster.local/
<unresolved>:30075,
```

```
resourcePath=/soothsayer/v1/canaryrelease/n5g-eir-eic,
authenticationType=OAUTH
```

- User accessing NRF resource without having NRF_READ role

```
logType=AUDIT, type=ACCESS_RESOURCE_ERROR, resourceType=NRF,
  userId=186f6f2a-ba6a-4812-8a18-b906a5f9e3f6, username=shreb,
status=403 FORBIDDEN, operationType=GET,
remoteAddress=/192.168.219.64:53587,
localAddress=cncc-core-ingress-gateway.cncc.svc.cluster.local/
<unresolved>:30075,
resourcePath=/nrf-configuration/v1/system-options,
authenticationType=OAUTH
```

- User successful logout

```
logType=AUDIT, type=LOGOUT, resourceType=CNCC,
userId=186f6f2a-ba6a-4812-8a18-b906a5f9e3f6, username=shreb,
operationType=POST, remoteAddress=/192.168.219.64:53587,
localAddress=cncc-core-ingress-gateway.cncc.svc.cluster.local/
<unresolved>:30075,
resourcePath=/logout, authenticationType=OAUTH
```

CNCC IAM:

- Login Error when password entered was wrong

```
04:56:35,890 WARN [org.keycloak.events] (default task-22)
                                     type=LOGIN_ERROR, realmId=master,
clientId=security-admin-console,
                                     userId=d7cde46f-15e1-4ff8-a2cb-
c5825e481438, ipAddress=192.168.219.64,
                                     error=invalid_user_credentials,
auth_method=openid-connect,
                                     auth_type=code, redirect_uri=http://
10.75.225.28:31373/cncc/auth/admin/master/console/, code_id=5aca4960-
eecf-406b-a7eb-92e249c2beeb,
                                     username=admin,
                                     authSessionParentId=5aca4960-eecf-406b-
a7eb-92e249c2beeb,
                                     authSessionTabId=8sruELA1WWs
```

- Login with correct credential

```
04:57:24,581 INFO [org.keycloak.events] (default task-22)
                                     type=LOGIN, realmId=master,
clientId=security-admin-console,
                                     userId=d7cde46f-15e1-4ff8-a2cb-
c5825e481438, ipAddress=192.168.219.64,
                                     auth_method=openid-connect, auth_type=code,
redirect_uri=http://10.75.225.28:31373/cncc/auth/admin/master/console/,
consent=no_consent_required,
                                     code_id=5aca4960-eecf-406b-
```

```
a7eb-92e249c2beeb, username=admin,  
                                authSessionParentId=5aca4960-  
eecf-406b-a7eb-92e249c2beeb,  
                                authSessionTabId=8sruELA1WWs
```

- **User created**

```
04:58:41,804 INFO [org.keycloak.events] (default task-22)  
                                operationType=CREATE, realmId=master,  
                                clientId=819ce4a5-ddbd-4717-908f-  
a204bdabc808,  
                                userId=d7cde46f-15e1-4ff8-a2cb-  
c5825e481438, ipAddress=192.168.219.64,  
                                resourceType=USER,  
                                resourcePath=users/070911f5-c397-42c1-  
b5a4-cd92fa435a33
```

- **Deleted user**

```
05:00:08,226 INFO [org.keycloak.events] (default task-22)  
                                operationType=DELETE, realmId=master,  
                                clientId=819ce4a5-ddbd-4717-908f-  
a204bdabc808,  
                                userId=d7cde46f-15e1-4ff8-a2cb-  
c5825e481438, ipAddress=192.168.219.64,  
                                resourceType=USER,  
                                resourcePath=users/  
2b931bbb-7f97-4f04-9f75-e0d0974ab73d
```

- **Admin Role removed for a user**

```
05:01:07,781 INFO [org.keycloak.events] (default task-22)  
                                operationType=DELETE, realmId=master,  
                                clientId=819ce4a5-ddbd-4717-908f-  
a204bdabc808,  
                                userId=d7cde46f-15e1-4ff8-a2cb-  
c5825e481438, ipAddress=192.168.219.64,  
                                resourceType=REALM_ROLE_MAPPING,  
                                resourcePath=users/  
08fc0058-133b-4288-9165-14c96c5dcd7a/role-mappings/realm
```

- **Admin Role added for a user**

```
05:01:33,664 INFO [org.keycloak.events] (default task-27)  
                                operationType=CREATE, realmId=master,  
                                clientId=819ce4a5-ddbd-4717-908f-  
a204bdabc808,  
                                userId=d7cde46f-15e1-4ff8-a2cb-  
c5825e481438, ipAddress=192.168.219.64,  
                                resourceType=REALM_ROLE_MAPPING,  
                                resourcePath=users/  
08fc0058-133b-4288-9165-14c96c5dcd7a/role-mappings/realm
```

- Realm setting update

```
05:02:29,222 INFO [org.keycloak.events] (default task-26)
                                operationType=UPDATE, realmId=master,
                                clientId=819ce4a5-ddbd-4717-908f-
a204bdabc808,
                                userId=d7cde46f-15e1-4ff8-a2cb-
c5825e481438, ipAddress=192.168.219.64,
                                resourceType=REALM, resourcePath=null
```

- Logout all session on keycloak

```
05:05:02,383 INFO [org.keycloak.events] (default task-29)
                                operationType=ACTION, realmId=master,
                                clientId=819ce4a5-ddbd-4717-908f-
a204bdabc808,
                                userId=d7cde46f-15e1-4ff8-a2cb-
c5825e481438, ipAddress=192.168.219.64,
                                resourceType=REALM, resourcePath=logout-all
```

Examples of Security Logs

Representation for IAM and Core are same as these logs are part of ingress-gateway. Only message part of the JSON log is shown in the example.

CNCC Core

- SCP request

```
logType=SECURITY, type=REQUEST, resourceType=SCP,
                                userId=3314f54f-08bf-489d-
b395-27bf56da1262, username=shreb,
                                operationType=GET, scheme=http,
                                remoteAddress=/192.168.219.64:53587,
                                localAddress=cncc-core-ingress-
gateway.cncc.svc.cluster.local/<unresolved>:30075,
                                resourcePath=/soothsayer/v1/
canaryrelease/, queryParams={},
                                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},
                                payload={}, authenticationType=OAUTH
```

- SCP response

```
logType=SECURITY, type=RESPONSE, resourceType=SCP,
                                userId=3314f54f-08bf-489d-
b395-27bf56da1262, username=shreb,
                                status=200 OK,
operationType=GET, scheme=http,
                                resourcePath=/soothsayer/v1/
canaryrelease/,
                                headers={transfer-
encoding=chunked, Connection=keep-alive,
                                Transfer-Encoding=chunked,
Content-Type=application/json,
                                Date=Sat, 04 Jul 2020 11:58:20
GMT},
                                payload=[{"serviceName":"n5g-
eir-
eic", "canaryReleaseFlag":true, "apiFullVersion":"2.0.0", "canaryTraffi
c":5}, {"serviceName":"namf-
comm", "canaryReleaseFlag":true, "apiFullVersion":"2.0.0", "canaryTraffi
c":5}, {"serviceName":"namf-
evts", "canaryReleaseFlag":true, "apiFullVersion":"2.0.0", "canaryTraffi
c":5}, {"serviceName":"namf-
loc", "canaryReleaseFlag":true, "apiFullVersion":"2.0.0", "canaryTraffi
c":5}, {"serviceName":"namf-
mt", "canaryReleaseFlag":true, "apiFullVersion":"2.0.0", "canaryTraffic
":5}, {"serviceName":"nausf-
auth", "canaryReleaseFlag":true, "apiFullVersion":"2.0.0", "canaryTraffi
c":5}, {"serviceName":"nausf-
sorprotection", "canaryReleaseFlag":true, "apiFullVersion":"2.0.0", "ca
naryTraffic":5}]],
                                authenticationType=OAUTH
```

CNCC IAM

- Request

```
logType=SECURITY, type=REQUEST, operationType=GET, scheme=http,
                                remoteAddress=/
192.168.219.64:53587,
                                localAddress=cncc-iam-ingress-
gateway.cncc.svc.cluster.local/<unresolved>:30085,
                                resourcePath=/cncc/auth/admin/
master/console/config,
                                queryParams={},
headers={Accept=application/json,
                                User-Agent=Mozilla/5.0 (Windows
NT 10.0; WOW64; rv:68.0)
                                Gecko/20100101 Firefox/68.0,
Referer=http://cncc-iam-ingress-
gateway.cncc.svc.cluster.local:30085/cncc/auth/admin/master/
console/, Connection=keep-alive,
                                Host=cncc-iam-ingress-
gateway.cncc.svc.cluster.local:30085,
                                Accept-Language=en-US,en;q=0.5,
```

```
Accept-Encoding=gzip, deflate,
                                DNT=1}, payload={},
                                authenticationType=NONE
```

- **Response**

```
logType=SECURITY, type=RESPONSE, status=200 OK,
                                operationType=GET, scheme=http,
                                resourcePath=/cncc/auth/admin/master/
console/config,
                                headers={transfer-encoding=chunked,
Cache-Control=no-cache,
                                X-XSS-Protection=1; mode=block, X-
Frame-Options=SAMEORIGIN,
                                Date=Mon, 06 Jul 2020 10:54:16 GMT,
Connection=keep-alive,
                                Strict-Transport-Security=max-
age=31536000; includeSubDomains,
                                X-Content-Type-Options=nosniff,
Content-Type=application/json,
                                Content-Length=211},

payload={"realm\":\"master\", \"auth-server-url\":\"http://cncc-iam-
ingress-gateway.cncc.svc.cluster.local:30085/cncc/auth/\", \"ssl-
required\":\"none\", \"resource\":\"security-admin-console\", \"public-
client\":true, \"confidential-port\":0},
                                authenticationType=NONE
```

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 10-1 Log View

