

Oracle® Communications

Cloud Native Configuration Console User Guide



Release 23.4.4
F87096-05
December 2024

ORACLE®

Copyright © 2019, 2024, 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, MySQL, and NetSuite 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	
1.1	CNC Console Compatibility Matrix	1
1.2	Reference	3
2	CNC Console Features	
2.1	Support for Network Functions	1
2.2	Support for CNE Common Services	7
2.3	LDAP Integration	7
2.4	SAML 2.0 Integration	9
2.5	Logging Support	9
2.6	Support for Multi Cluster Deployment	11
2.7	cnDBTier Integration	13
2.8	TLS Certificate Support	13
2.9	Service Mesh Communication	14
2.10	Support for IPv4 or IPv6	14
2.11	CNC Console GUI Session Timeout	14
2.12	Support for Kubernetes Resource	15
2.12.1	Network Policies	15
3	Logging into CNC Console	
3.1	CNC Console Dashboard	2
4	Configuring CNC Console IAM	
4.1	Role Based Access Control in CNC Console IAM	1
4.1.1	Types of Roles in CNC Console	2
4.1.2	Accessing Roles in CNC Console Applications	2
4.1.3	Updating Admin Password in CNC Console IAM	3
4.1.4	Creating or Updating User Password in CNC Console IAM	6
4.1.5	Password Policies for CNC Console Users	7
4.2	Configuring the CNC Console Redirection URL	8
4.3	Users in CNC Console IAM	9

4.3.1	Creating the Users	9
4.3.2	Viewing the Users	12
4.3.3	Assigning Roles to the User	12
4.4	Integrating SAML SSO with CNC Console IAM	14
4.5	Integrating CNC Console LDAP Server with CNC Console IAM	22
4.5.1	Configuring User Federation with CNC Console IAM	23
4.5.2	Grouping the LDAP Mapper and Assigning the Roles	27

5 Accessing NF Configurations through CNC Console IAM

5.1	Generate Access Tokens	1
5.2	Refresh Access Tokens	3
5.3	Accessing NF Resources	4

6 CNC Console Metrics

6.1	CNC Console IAM Metrics	2
6.1.1	M-CNCC IAM Requests	2
6.1.2	M-CNCC IAM Response	2
6.1.3	M-CNCC IAM Success Responses	3
6.1.4	M-CNCC IAM 5xx Responses	3
6.1.5	M-CNCC IAM 4xx Responses	4
6.1.6	M-CNCC IAM Error Responses	4
6.1.7	M-CNCC IAM Access Token Request	5
6.1.8	M-CNCC IAM Access Token Granted	6
6.1.9	M-CNCC IAM Access Token Not Granted	6
6.1.10	M-CNCC IAM User Login Failure Responses	7
6.2	M-CNCC Core Metrics	8
6.2.1	M-CNCC Core Requests	8
6.2.2	M-CNCC Core Responses	9
6.2.3	M-CNCC Core Success Responses	9
6.2.4	M-CNCC Core 5xx Responses	10
6.2.5	M-CNCC Core 4xx Responses	10
6.2.6	M-CNCC Core Error Responses	11
6.2.7	M-CNCC Core Access Token Request	11
6.2.8	M-CNCC Core Access Token Granted Responses	12
6.2.9	M-CNCC Core Access Token Not Granted Responses	13
6.2.10	M-CNCC Core User Login Failure Responses	14
6.2.11	M-CNCC Core User Authorization Failure Responses	15
6.2.12	M-CNCC Core BSF Requests	16
6.2.13	M-CNCC Core BSF Responses	17
6.2.14	M-CNCC Core DD Requests	17

6.2.15	M-CNCC Core DD Responses	18
6.2.16	M-CNCC Core PROVGW Requests	18
6.2.17	M-CNCC Core PROVGW Responses	19
6.2.18	M-CNCC Core NRF Responses	19
6.2.19	M-CNCC Core NRF Requests	20
6.2.20	M-CNCC Core NSSF Requests	21
6.2.21	M-CNCC Core NSSF Responses	21
6.2.22	M-CNCC Core NWDAF Requests	21
6.2.23	M-CNCC Core NWDAF Responses	22
6.2.24	M-CNCC Core POLICY Requests	22
6.2.25	M-CNCC Core POLICY Responses	23
6.2.26	M-CNCC Core SCP Responses	23
6.2.27	M-CNCC Core SCP Requests	24
6.2.28	M-CNCC Core SEPP Requests	24
6.2.29	M-CNCC Core SEPP Responses	25
6.2.30	M-CNCC Core UDR Requests	25
6.2.31	M-CNCC Core UDR Responses	26
6.2.32	M-CNCC Core cnDBTier Requests	27
6.2.33	M-CNCC Core cnDBTier Responses	27
6.2.34	M-CNCC Core OCCM Requests	27
6.2.35	M-CNCC Core OCCM Responses	28
6.3	A-CNCC Core Metrics	28
6.3.1	A-CNCC Core Requests	28
6.3.2	A-CNCC Core Responses	29
6.3.3	A-CNCC Core Success Responses	29
6.3.4	A-CNCC Core 5xx Responses	30
6.3.5	A-CNCC Core 4xx Responses	30
6.3.6	A-CNCC Core Error Responses	31
6.3.7	A-CNCC Core Access Token Request	31
6.3.8	A-CNCC Core Access Token Granted Responses	32
6.3.9	A-CNCC Core Access Token Not Granted Responses	33
6.3.10	A-CNCC Core User Authorization Failure Responses	34
6.3.11	A-CNCC Core SCP Requests	35
6.3.12	A-CNCC Core SCP Responses	36
6.3.13	A-CNCC Core NRF Requests	36
6.3.14	A-CNCC Core NRF Responses	37
6.3.15	A-CNCC Core UDR Requests	37
6.3.16	A-CNCC Core UDR Responses	38
6.3.17	A-CNCC Core POLICY Requests	38
6.3.18	A-CNCC Core POLICY Responses	39
6.3.19	A-CNCC Core BSF Requests	40
6.3.20	A-CNCC Core BSF Responses	40

6.3.21	A-CNCC Core SEPP Requests	41
6.3.22	A-CNCC Core SEPP Responses	41
6.3.23	A-CNCC Core NSSF Requests	42
6.3.24	A-CNCC Core NSSF Responses	42
6.3.25	A-CNCC Core DD Requests	42
6.3.26	A-CNCC Core DD Responses	43
6.3.27	A-CNCC Core PROVGW Requests	43
6.3.28	A-CNCC Core PROVGW Responses	44
6.3.29	A-CNCC Core NWDAF Requests	44
6.3.30	A-CNCC Core NWDAF Responses	45
6.3.31	A-CNCC Core cnDBTier Requests	45
6.3.32	A-CNCC Core cnDBTier Responses	46
6.3.33	A-CNCC Core OCCM Requests	46
6.3.34	A-CNCC Core OCCM Responses	47

7 CNC Console Alerts

7.1	CNC Console IAM Alerts	1
7.1.1	CncclamTotalIngressTrafficRateAboveMinorThreshold	1
7.1.2	CncclamTotalIngressTrafficRateAboveMajorThreshold	2
7.1.3	CncclamTotalIngressTrafficRateAboveCriticalThreshold	3
7.1.4	CncclamMemoryUsageCrossedMinorThreshold	4
7.1.5	CncclamMemoryUsageCrossedMajorThreshold	4
7.1.6	CncclamMemoryUsageCrossedCriticalThreshold	5
7.1.7	CncclamTransactionErrorRateAbove0.1Percent	5
7.1.8	CncclamTransactionErrorRateAbove1Percent	6
7.1.9	CncclamTransactionErrorRateAbove10Percent	6
7.1.10	CncclamTransactionErrorRateAbove25Percent	7
7.1.11	CncclamTransactionErrorRateAbove50Percent	7
7.1.12	CncclamIngressGatewayServiceDown	8
7.1.13	CncclamFailedLogin	9
7.1.14	AdminUserCreation	9
7.1.15	CncclamAccessTokenFailure	10
7.2	CNC Console Core Alerts	10
7.2.1	CnccCoreTotalIngressTrafficRateAboveMinorThreshold	11
7.2.2	CnccCoreTotalIngressTrafficRateAboveMajorThreshold	11
7.2.3	CnccCoreTotalIngressTrafficRateAboveCriticalThreshold	12
7.2.4	CnccCoreMemoryUsageCrossedMinorThreshold	13
7.2.5	CnccCoreMemoryUsageCrossedMajorThreshold	13
7.2.6	CnccCoreMemoryUsageCrossedCriticalThreshold	14
7.2.7	CnccCoreTransactionErrorRateAbove0.1Percent	15
7.2.8	CnccCoreTransactionErrorRateAbove1Percent	15

7.2.9	CnccCoreTransactionErrorRateAbove10Percent	16
7.2.10	CnccCoreTransactionErrorRateAbove25Percent	16
7.2.11	CnccCoreTransactionErrorRateAbove50Percent	17
7.2.12	CnccCoreIngressGatewayServiceDown	17
7.2.13	CnccCoreFailedLogin	18
7.2.14	CnccCoreUnauthorizedAccess	18
7.2.15	CnccCoreAccessTokenFailure	19
7.3	CNCC Alert configuration in Prometheus	20
7.4	Validating Alerts	21
7.5	Disabling Alerts	22
7.6	Configuring SNMP-Notifier	22
7.7	MIB Files for CNCC	23

8 CNC Console KPIs

8.1	CNC Console IAM KPIs	1
8.1.1	M-CNCC IAM Requests	1
8.1.2	M-CNCC IAM Responses	1
8.1.3	M-CNCC IAM Success Rate	2
8.1.4	M-CNCC IAM Error Rate	3
8.2	M-CNCC Core KPIs	3
8.2.1	M-CNCC Core Requests	3
8.2.2	M-CNCC Core Responses	5
8.2.3	M-CNCC Core Success Rate	8
8.2.4	M-CNCC Core Error Rate	13
8.3	A-CNCC Core KPIs	18
8.3.1	A-CNCC Core Requests	18
8.3.2	A-CNCC Core Responses	20
8.3.3	A-CNCC Core Success Rate	23
8.3.4	A-CNCC Core Error Rate	28
8.4	CNC Console Common KPIs	33
8.4.1	CPU Usage	33
8.4.2	Memory Usage	34

9 CNC Console Logs

9.1	Log Formats	1
9.2	Types of Logs	4
9.3	Examples of Logs	8
9.4	Accessing logs	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 terminologies used in the document:

Table Acronyms

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

Table (Cont.) Acronyms

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

What's New in This Guide

This section introduces the documentation updates for release 23.4.x.

Release 23.4.4 - F87096-04, December 2024

There are no updates in this release.

Release 23.4.3 - F87096-03, October 2024

- Updated the [Compatibility Matrix](#) section to provide information on CNC Console compatibility with latest NF versions.
- Added the Updating M-CNCC IAM KC log level to the [Types of Logs](#) section.

Release 23.4.2 - F87096-03, July 2024

- Updated the [Compatibility Matrix](#) section to provide information on CNC Console compatibility with latest NF versions.

Release 23.4.1 - F87096-02, April 2024

Updated the metrics list in the [Memory Usage](#) KPI.

Release 23.4.0 - F87096-01, December 2023

- The metrics section is rearranged and the [M-CNCC Core Metrics](#) section is updated with the following metrics:
 - M-CNCC Core Requests
 - M-CNCC Core Responses
 - M-CNCC Core Success Responses
 - M-CNCC Core 5xx Responses
 - M-CNCC Core 4xx Responses
 - M-CNCC Core Error Responses
 - M-CNCC Core Access Token Request
 - M-CNCC Core Access Token Granted Responses
 - M-CNCC Core Access Token Not Granted Responses
 - M-CNCC Core User Login Failure Responses
 - M-CNCC Core User Authorization Failure Responses
 - M-CNCC Core SCP Requests
 - M-CNCC Core NRF Requests
 - M-CNCC Core UDR Requests
 - M-CNCC Core POLICY Requests
 - M-CNCC Core BSF Requests
 - M-CNCC Core SEPP Requests
 - M-CNCC Core NSSF Requests
 - M-CNCC Core DD Requests
 - M-CNCC Core PROVGW Requests
 - M-CNCC Core NWDAF Requests

-
- M-CNCC Core cnDBTier Requests
 - M-CNCC Core OCCM Requests
 - M-CNCC Core SCP Responses
 - M-CNCC Core NRF Responses
 - M-CNCC Core UDR Responses
 - M-CNCC Core POLICY Responses
 - M-CNCC Core BSF Responses
 - M-CNCC Core SEPP Responses
 - M-CNCC Core NSSF Responses
 - M-CNCC Core DD Responses
 - M-CNCC Core PROVGW Responses
 - M-CNCC Core NWDAF Responses
 - M-CNCC Core cnDBTier Responses
 - M-CNCC Core OCCM Responses
 - The metrics section is rearranged and the [A-CNCC Core Metrics](#) section is added with the following metrics:
 - A-CNCC Core Requests
 - A-CNCC Core Responses
 - A-CNCC Core Success Responses
 - A-CNCC Core 5xx Responses
 - A-CNCC Core 4xx Responses
 - A-CNCC Core Error Responses
 - A-CNCC Core Request Processing Time-Ingress only
 - A-CNCC Core Request Processing Time
 - A-CNCC Core Access Token Request
 - A-CNCC Core Access Token Granted Responses
 - A-CNCC Core Access Token Not Granted Responses
 - A-CNCC Core User Authorization Failure Responses
 - A-CNCC Core SCP Requests
 - A-CNCC Core NRF Requests
 - A-CNCC Core UDR Requests
 - A-CNCC Core POLICY Requests
 - A-CNCC Core BSF Requests
 - A-CNCC Core SEPP Requests
 - A-CNCC Core NSSF Requests
 - A-CNCC Core DD Requests
 - A-CNCC Core PROVGW Requests
 - A-CNCC Core NWDAF Requests

- A-CNCC Core cnDBTier Requests
- A-CNCC Core OCCM Requests
- A-CNCC Core SCP Responses
- A-CNCC Core NRF Responses
- A-CNCC Core UDR Responses
- A-CNCC Core POLICY Responses
- A-CNCC Core BSF Responses
- A-CNCC Core SEPP Responses
- A-CNCC Core NSSF Responses
- A-CNCC Core DD Responses
- A-CNCC Core PROVGW Responses
- A-CNCC Core NWDAF Responses
- A-CNCC Core cnDBTier Responses
- A-CNCC Core OCCM Responses
- The KPIs section is rearranged and the [M-CNCC Core KPIs](#) is updated with section with the following KPIs:
 - M-CNCC Core Requests
 - M-CNCC Core Responses
 - M-CNCC Core Success Rate
 - M-CNCC Core Error Rate
 - M-CNCC IAM Error Rate
- The KPIs section is rearranged and the [A-CNCC Core KPIs](#) section is added with the following KPIs:
 - A-CNCC Core Requests
 - A-CNCC Core Responses
 - A-CNCC Core Success Rate
 - A-CNCC Core Error Rate
- Added the [Password Policies for CNC Console Users](#) section describing all the password policies for CNC Console Core users.

1

Introduction

The Cloud Native Configuration 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.

1.1 CNC Console Compatibility Matrix

This section lists the versions of added or updated components in release 23.4.x. To know the list of all the supported versions, see Oracle Communications Cloud Native Core Release Notes.

Release 23.4.2

The following table lists the versions of added or updated components in release 23.4.2:

Table 1-1 Compatibility Matrix

Network Functions	Compatible Versions
BSF	23.4.x
NRF	23.4.x
NSSF	23.4.x
Policy	23.4.x
SCP	23.4.x
SEPP	23.4.x
UDR	23.4.x

CNC Console is compatible with the following components:

Table 1-2 Compatibility Matrix

Components	Compatible Versions
OCNADD	23.4.x
CNE	23.4.x, 23.3.x, 23.2.x
cnDBTier	23.4.x, 23.3.x, 23.2.x
CDCS	23.4.x, 23.3.x, 23.2.x
OSO	23.4.x, 23.3.x, 23.2.x

Table 1-2 (Cont.) Compatibility Matrix

Components	Compatible Versions
ASM	1.14.6-am1, 1.11.8-am1. 1.9.8-am1
OCNWDAF	23.4.x
PROVGW	23.4.x
OCCM	23.4.x

Release 23.4.1

The following table lists the versions of added or updated components in release 23.4.1:

Table 1-3 Compatibility Matrix

Network Functions	Compatible Versions
BSF	23.4.x
NRF	23.4.x
NSSF	23.4.x
Policy	23.4.x
SCP	23.4.x
SEPP	23.4.x
UDR	23.4.x

CNC Console is compatible with the following components:

Table 1-4 Compatibility Matrix

Components	Compatible Versions
OCNADD	23.4.x
CNE	23.4.x, 23.3.x, 23.2.x
cnDBTier	23.4.x, 23.3.x, 23.2.x
CDCS	23.4.x, 23.3.x, 23.2.x
OSO	23.4.x, 23.3.x, 23.2.x
ASM	1.14.6-am1, 1.11.8-am1. 1.9.8-am1
OCNWDAF	23.4.x
PROVGW	23.4.x
OCCM	23.4.x

Release 23.4.0

The following table lists the versions of added or updated components in release 23.4.0:

Table 1-5 Compatibility Matrix

Network Functions	Compatible Versions
BSF	23.4.x
NRF	23.4.x
NSSF	23.4.x

Table 1-5 (Cont.) Compatibility Matrix

Network Functions	Compatible Versions
Policy	23.4.x
SCP	23.4.x
SEPP	23.4.x
UDR	23.4.x

CNC Console is compatible with the following components:

Table 1-6 Compatibility Matrix

Components	Compatible Versions
OCNADD	23.4.x
CNE	23.4.x, 23.3.x, 23.2.x
cnDBTier	23.4.x, 23.3.x, 23.2.x
CDCS	23.4.x, 23.3.x, 23.2.x
OSO	23.3.x, 23.2.x, 22.3.x
ASM	1.14.6-am1, 1.11.8-am1, 1.9.8-am1
OCNWDAP	23.4.x
PROVGW	23.4.x
OCCM	23.4.x

1.2 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 Slice Selection Function User Guide*
- *Oracle Communications Cloud Native Core, Network Repository Function Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Service Communication Proxy Installation Guide*
- *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery 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*

- *Oracle Communications Cloud Native Core, Network Slice Selection Function Installation Guide*
- *Oracle Communications Networks Data Analytics Installation and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Certificate Management User Guide*
- *Oracle Communications Cloud Native Core, Certificate Management Installation, Upgrade, and Fault Recovery Guide*

2

CNC Console Features

This section describes the features supported by CNC Console.

2.1 Support for Network Functions

CNC Console GUI provides a user interface to configure and manage the following Network Functions (NFs):

Oracle Communications Cloud Native Core, Binding Support Function (BSF)

BSF provides the following functions:

- Allows PCF users to register, update, and remove the binding information
- Allows NF 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*.

Note

: The performance and capacity of the CNC Console system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment.

Oracle Communications Network Analytics Data Director (OCNADD)

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

Data collection is a complex task in a 5G Service Based Architecture (SBA). The data collected is to provide meaningful insights to the customers. The OCNADD can filter, replicate, aggregate data and route these data feeds to third party consumers who have subscribed to the feeds. OCNADD ensures data security, low latency, and redundancy while collecting and processing the data feeds. The OCNADD 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). This enables them to 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 OCNADD generated data feed can be beneficial for monitoring and troubleshooting during a network failure. OCNADD is a crucial function that aids in creating self-healing networks.

Oracle Communications Cloud Native Core, Network Repository Function (NRF)

NRF 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 NRF 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*.

Oracle Communications Cloud Native Core, Network Slice Selection Function (NSSF)

NSSF provides the following functions:

Network slices allow 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 NF 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 NSSF fulfills the requirement for determining the individual NF pertaining to a slice.

NSSF is a functional element that supports the following functionalities:

- NSSF 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.
- NSSF uses an NF Service Consumer (AMF) to update the S-NSSAIs that AMF supports and notifies of any changes in the status.
- NSSF selects the network slicing instance (NSI) and determines the authorized Network Slice Selection Assistance Information (NSSAIs) and AMF to serve the UE.
- NSSF interaction with NRF allows retrieving specific NF services to be used for registration request.

NSSF 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)

NSSF 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.

Oracle Communications Networks Data Analytics Function (NWDAF)

The NWDAF enables the operator to collect and analyze the data in the network through an analytics function. The 5G technology requires prescriptive analytics to drive closed-loop automation and self-healing networks. In a 5G network, the consumers of data are 5G NFs, Application Functions (AFs), and Operations, Administration, and Maintenance (OAM) and the data producers are NFs. The NWDAF supports the following functions:

- NWDAF collects data from Access and Mobility Management Function (AMF), Session Management Function (SMF), and Network Repository Function (NRF) in the network. The data is collected directly from the NFs or through the Network Exposure Function (OCNEF).
- The NWDAF is designed to provide analytics information to consumer such as NFs, AFs and OAM.

A 5G network contains a vast number of devices and sensors generating an enormous amount of data. The NWDAF function allows the Communications Service Providers (CSPs) to efficiently monitor, manage, automate, and optimize their network operations by the data collected and analytics generated across the network. The NWDAF also helps the CSPs in achieving the operational efficiency and provides an enhanced service experience.

The analytics information provided by the NWDAF is either statistical information based on past events or predictive information. This analytics information is used to balance the resources on the network. The NWDAF can predict the User Equipment (UE) location and also detect if the UE is in an abnormal location. Based on the collected analytics information, the CSPs can roll out new services or modify the existing services without waiting for a maintenance window in the network. This ensures significantly fewer chances of network experiencing downtime.

An NWDAF consumer can avail analytics information for different analytic events. Alternatively, the consumers can subscribe or unsubscribe for specific analytics information as a one-time event or periodically get notified when a specifically defined event (for example, a threshold is breached) is detected.

The NRF discovers the NWDAF instances for the NF consumers in the network. The NWDAF information can also be locally configured on the NF consumers. The NWDAF selection function in the consumer NF selects an NWDAF instance among available NWDAF instances. Different NWDAF instances present in the 5G network can be configured to provide a specific type of analytics information. This information about the NWDAF instance is described in the

NWDAF profile stored in the NRF. The consumer NFs that need specific analytics types query the NRF and include the Analytics ID based on the required data.

Oracle Communications Cloud Native Core, Converged Policy (Policy)

- Policy is an NF 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

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 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*.

Provisioning Gateway (PROVGW)

Oracle Provisioning Gateway for Subscriber Location Function (SLF) is implemented as a cloud native function. It supports:

- HTTP1.1 over TLS
- Custom entities or fields in UDR mode over SOAP/XML interface
- Conversion of requests as defined in SEC.yaml configurations for SOAP/XML interface in UDR mode
- Auditor functionality in SLF mode
- OAM interface and configuration APIs to configure Ingress Gateway, Egress Gateway, and other Provisioning Gateway microservices

It has two modes, which are as follows:

- SLF mode: This mode is applicable when Provisioning Gateway is deployed with UDR for SLF usecase. You can configure Provisioning Gateway to connect to multiple segments of SLF where each segment has two SLFs. This mode offers an HTTP2 based secured

REST/JSON interface (through Ingress Gateway API) for SLF data provisioning. It relays the request received by the provisioning client (e.g. MTAS) to multiple 5G UDR or SLF segments. The response received from each UDR or SLF segment is then consolidated and the final response is sent to the provisioning system.

You can deploy multiple Provisioning Gateways in the same segment or across multiple segments where, each one is stateless and does not interact with each other. If one of the Provisioning Gateway goes down, MTAS can use second Provisioning Gateway instance to continue provisioning SLF data on UDR.

- UDR mode: This mode is applicable when deployed with UDR for converged policy database solution. In this mode, Provisioning Gateway supports SOAP/XML interface, which is similar to 4G UDR.

Oracle Communications Cloud Native Core, 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*.

Oracle Communications Cloud Native Core, Security Edge Protection Proxy (SEPP)

SEPP 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*.

Oracle Communications Cloud Native Core, 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

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.

2.2 Support for CNE Common Services

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

When CNC Console is integrated with common services, it provides an additional layer of security through authentication and authorization for common services that don't have their own authentication mechanism. CNC Console also provides the user a single login for all common services as per your assigned roles.

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

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

Prometheus and Alertmanager GUIs can be accessed using the CNC Console. For more information on accessing Prometheus and Alertmanager GUIs using CNC Console, refer to *Oracle Communications Cloud Native Configuration Console User 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.

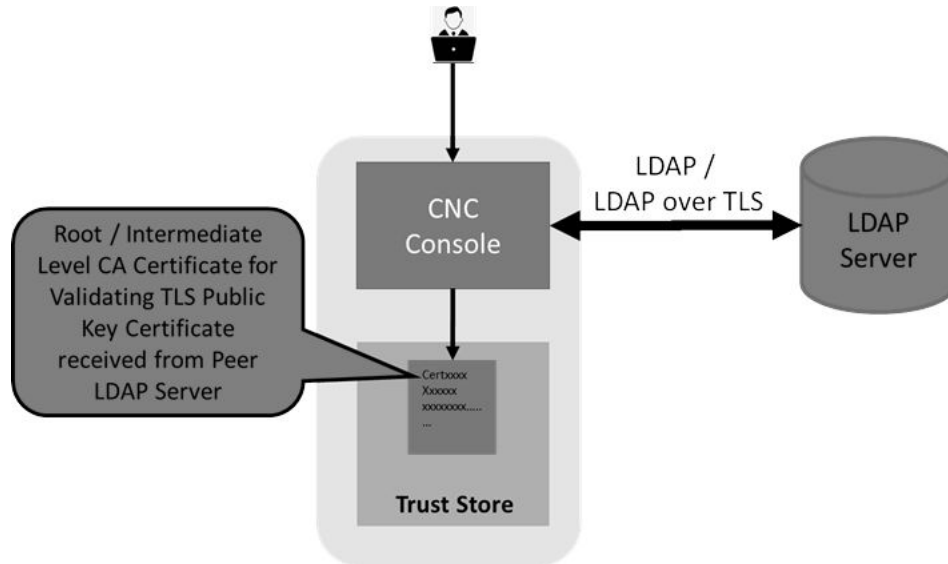
In case the alert still persists, perform the following:

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.

2.3 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.

2.4 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.

2.5 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.

2.6 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 Multicluster 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 multicluster 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

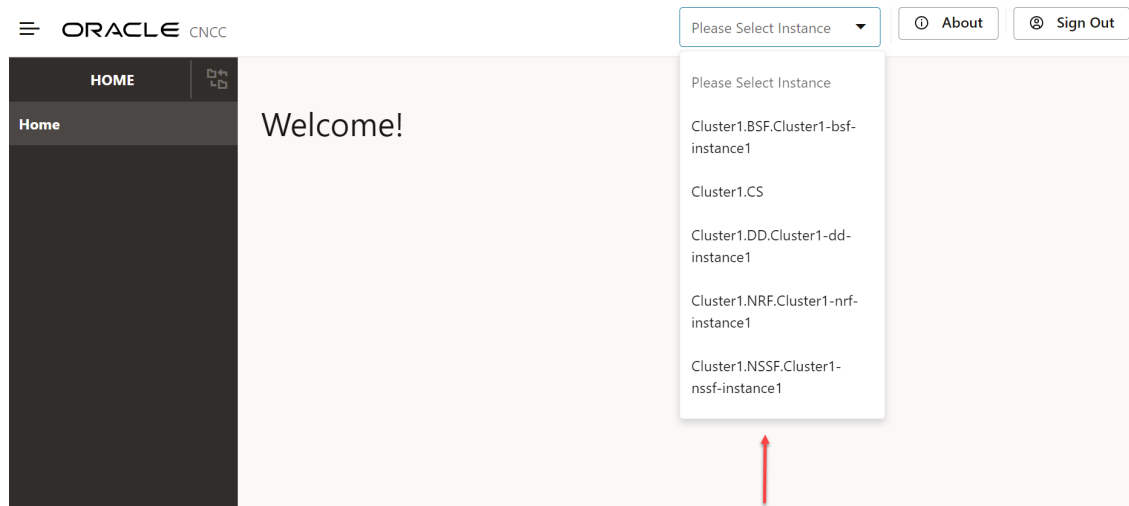
CNC Console supports multiple instances of NFs within a Kubernetes cluster using Manager CNC Console IAM (M-CNCC IAM), Manager CNC Console Core (M-CNCC Core), and Agent CNC Console Core (A-CNCC Core).

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 Configuration Console, Installation, Upgrade, and Fault Recovery Guide*.

2.7 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*.

2.8 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 provides secure and encrypted communication. It ensures that the information you send and receive, such as passwords, or personal data, is protected from being intercepted by malicious parties. HTTPS achieves this security through TLS (Transport Layer Security). CNC Console supports one-way TLS and mTLS (mutual TLS). With one-way TLS, only the website you're connecting to is verified, ensuring that your data is securely transmitted to the server. With mTLS, both your device and the website verify each others identities, adding an extra layer of trust and security to the communication. mTLS provides a stronger authentication process to ensure that you're securely connecting to the intended website and not falling victim to impersonation or man-in-the-middle attacks.

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.

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)
- Rivest, Shamir, Adleman (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

2.9 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 Configuration Console Installation and Upgrade Guide*.

2.10 Support for IPv4 or IPv6

CNC Console can be deployed on CNE that supports dual stack networking. Applications or NFs can establish connections with pods and services in a Kubernetes cluster using IPv4 or IPv6. CNC Console can be configured to use either IPv4 or IPv6. By default, IPv4 is used if no configuration is provided.

2.11 CNC Console GUI Session Timeout

The duration (in seconds) before a CNC Console GUI session times out and the user has to log in again can be configured using the `ingress-gateway.cncc.core.sessionTimeout` parameter in the `custom_values.yaml` file.

By default, the session time out value is set to 1800 seconds. However, you can set it to any value between 300 and 7200 seconds.

Note

The value of the CNC Console IAM SSO session idle timeout configuration is not considered for CNC Console Core session management.

2.12 Support for Kubernetes Resource

2.12.1 Network Policies

Network Policies are an application-centric construct that allows you to specify how a pod communicates with various network entities. It creates pod-level rules to control communication between the cluster pods and services, and to determine which pods and services can access one another inside a cluster.

Previously, the pods under CNC Console deployment can be contacted by any other pods in the Kubernetes cluster without any restrictions. Now, Network Policy provides namespace-level isolation, which allows secured communications to and from CNC Console with rules defined in respective Network Policies. The network policies enforce access restrictions for all the applicable data flows except communication from Kubernetes node to pod for invoking container probe. For example, CNC Console internal microservices cannot be contacted directly by any other pods.

Managing Support for Network Policies

Enable

To use this feature, network policies need to be applied to the namespace wherein CNC Console is applied.

Configure

You can configure this feature using Helm. For information about Configuring Network Policy for CNC Console Deployment, see *Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*.

Observe

There are no specific metrics and alerts required for the Support of Network Policy functionality.

3

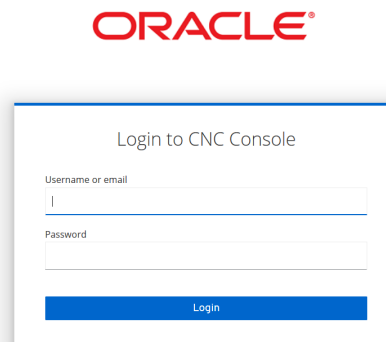
Logging into CNC Console

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

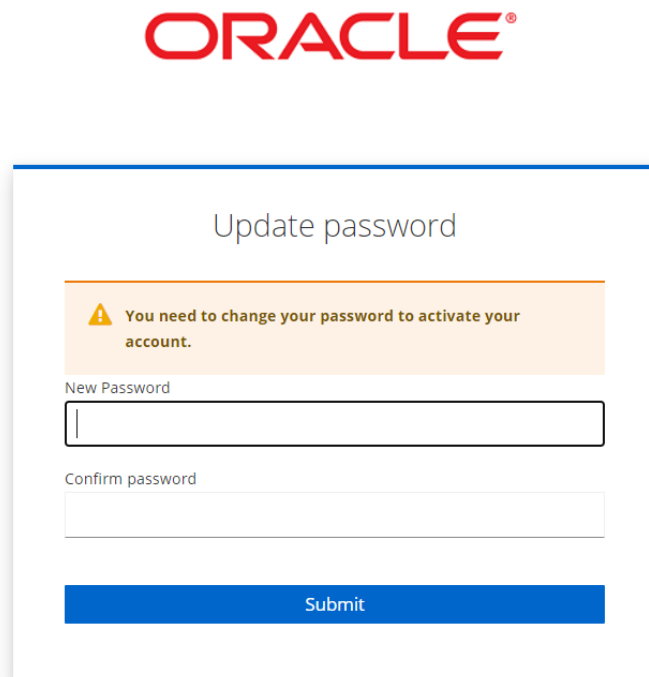
1. Open any browser.
2. Enter the URL: ***http://<host name>:<port number>***.
where, *host name* is cncc-mcore-ingress-ip and *port number* is cncc-mcore-ingressport.

The following screen appears:

Figure 3-1 CNC Console Login



3. Enter the valid credentials.
4. Click **Login**. The Welcome page of CNC Console interface appears.
5. After the first login, you will be prompted to change password.

Figure 3-2 Update Password

ORACLE®

Update password

⚠ You need to change your password to activate your account.

New Password

Confirm password

Submit

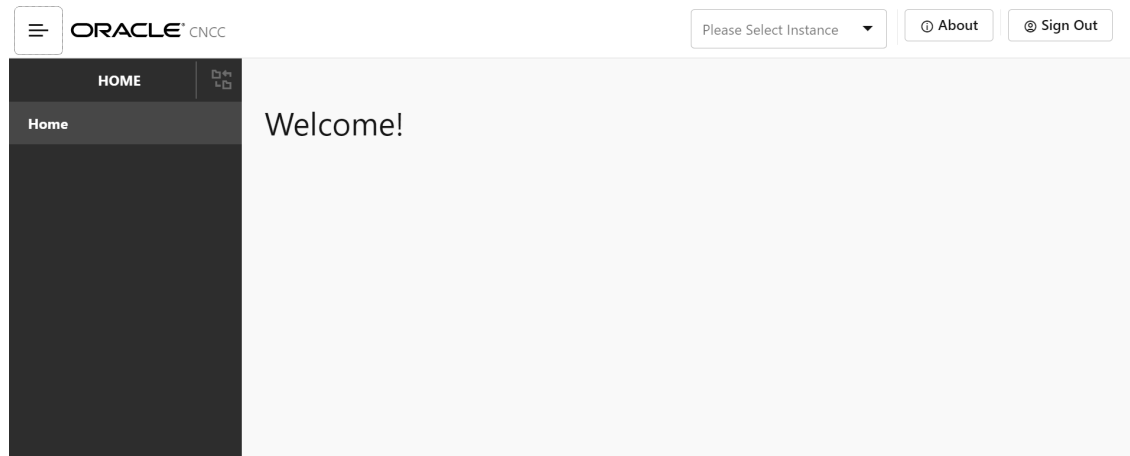
Note

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

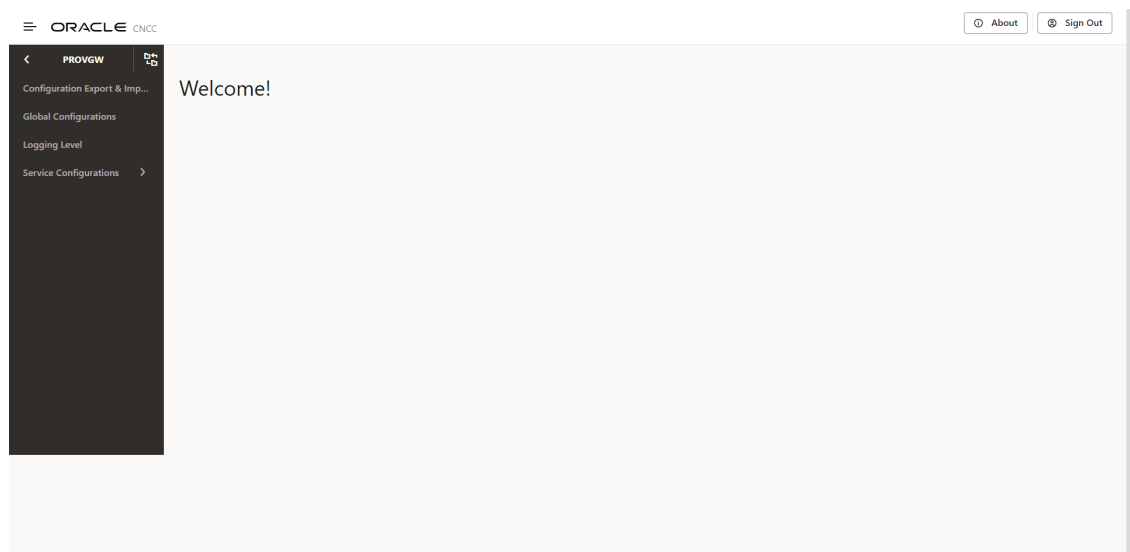
3.1 CNC Console Dashboard

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 3-3 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 3-4 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

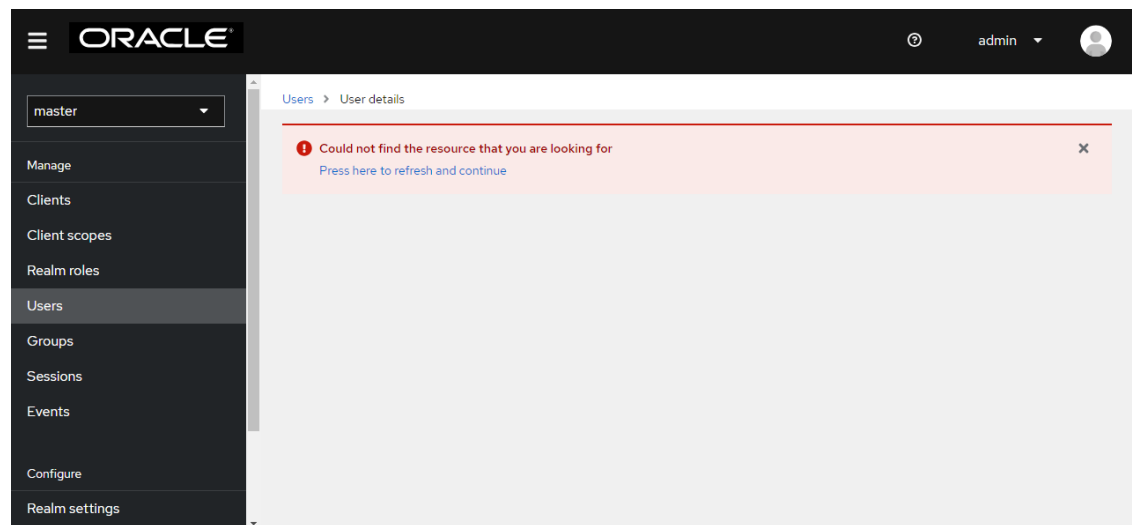
Configuring CNC Console IAM

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

Restricted Actions on CNC Console IAM

You can only perform those actions that are listed in the *Oracle Communications Cloud Native Configuration Console User Guide* using CNC Console IAM. The following error message appears if you try to perform a restricted action:

Figure 4-1 Restricted Actions Error Message



Click the **Press here to refresh and continue** link to reload CNC Console IAM.

4.1 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.

4.1.1 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.

 Note

CNC Console IAM is managed by an admin who is the single user. You can't have more than one admin user per IAM instance.

- **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**

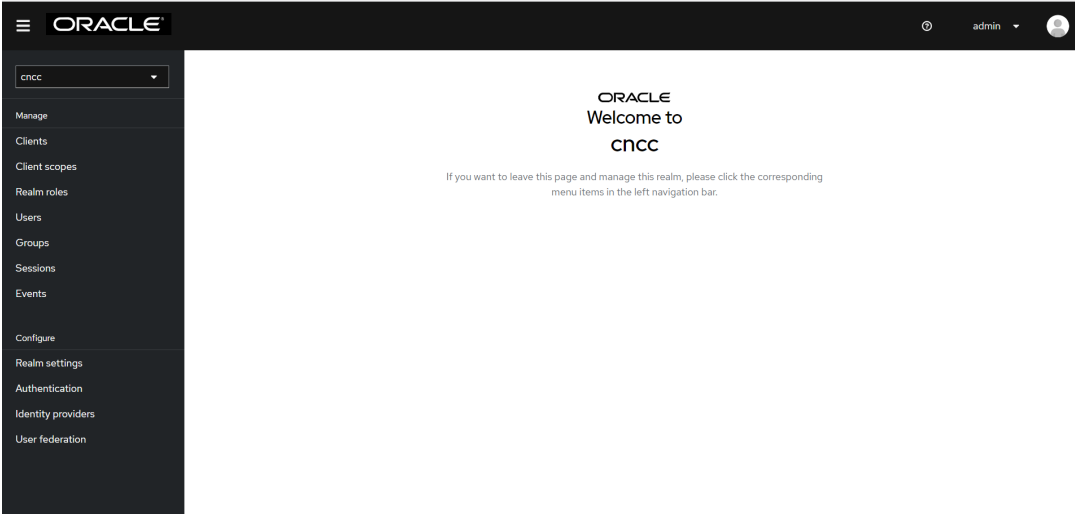
The user assigned with this role has access to all the common services and can perform CRUD operations.

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, Promxy, OpenSearch, and Jaeger-ES 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.

4.1.2 Accessing Roles in CNC Console Applications

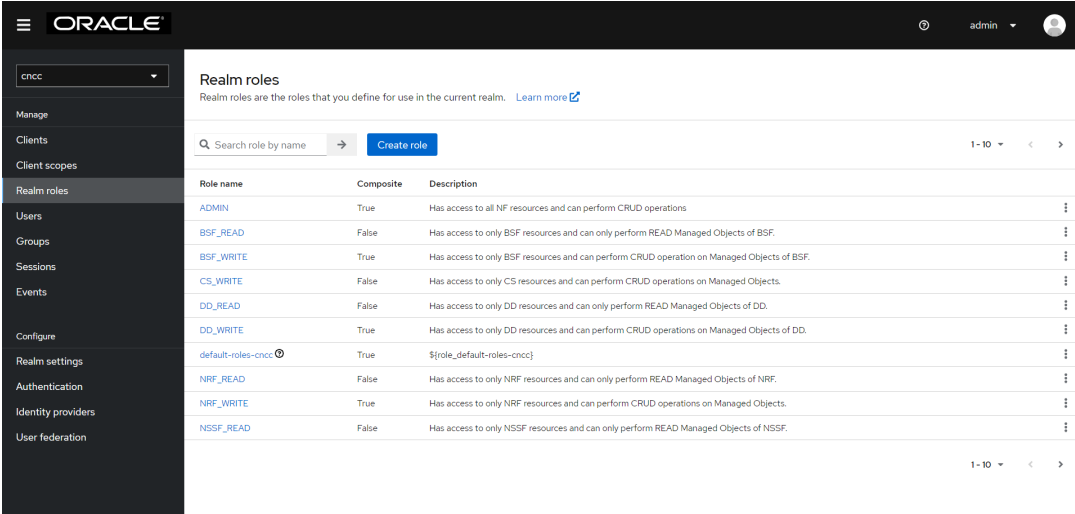
1. Log into CNC Console IAM using Admin credentials and select the CNCC Realm. The following screen appears.

Figure 4-2 Realm Settings



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

Figure 4-3 Realm Roles



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 Configuration Console Installation and Upgrade Guide*.

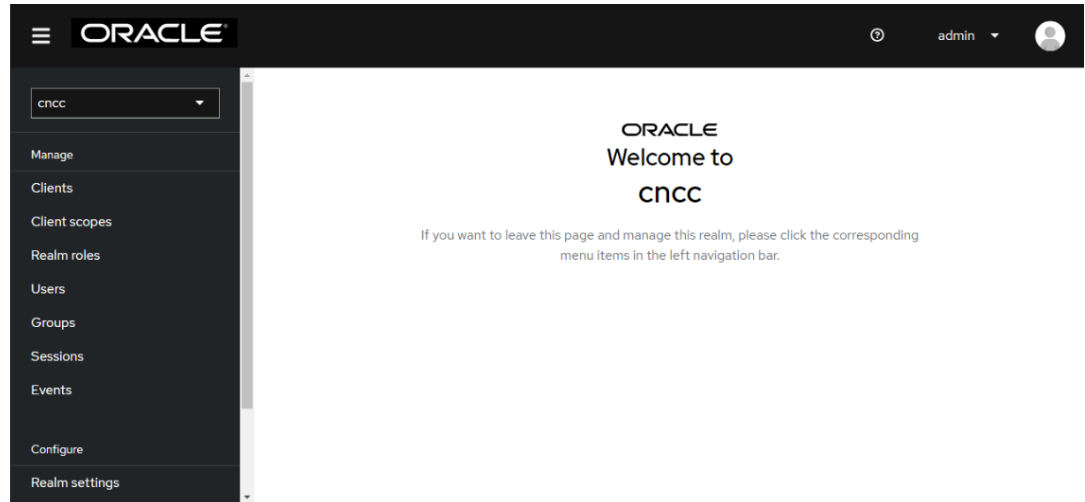
4.1.3 Updating Admin Password in CNC Console IAM

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

CNC Console provides support to change the password of CNC Console IAM. To update password:

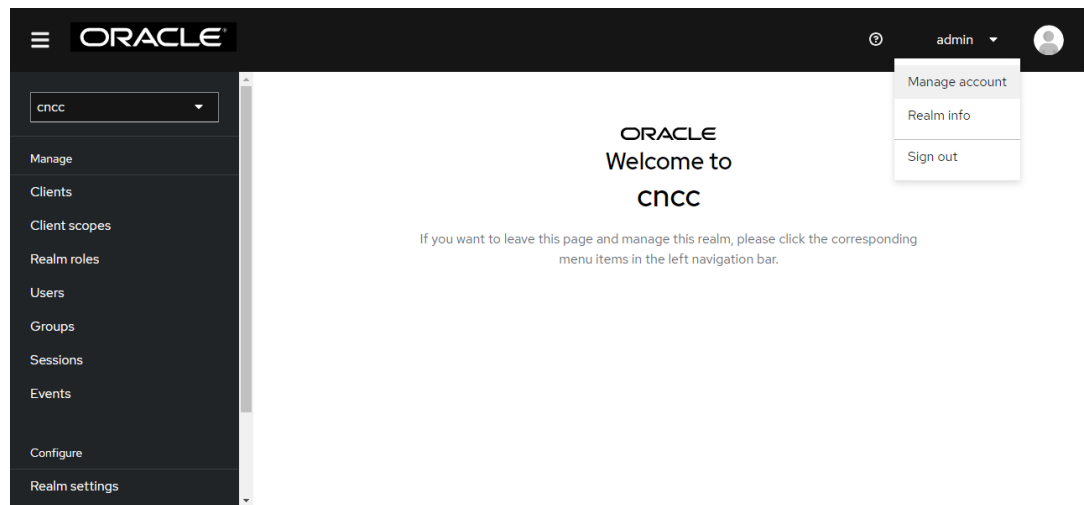
1. Login to CNC Console IAM and click **admin**.

Figure 4-4 Log in to CNC Console IAM

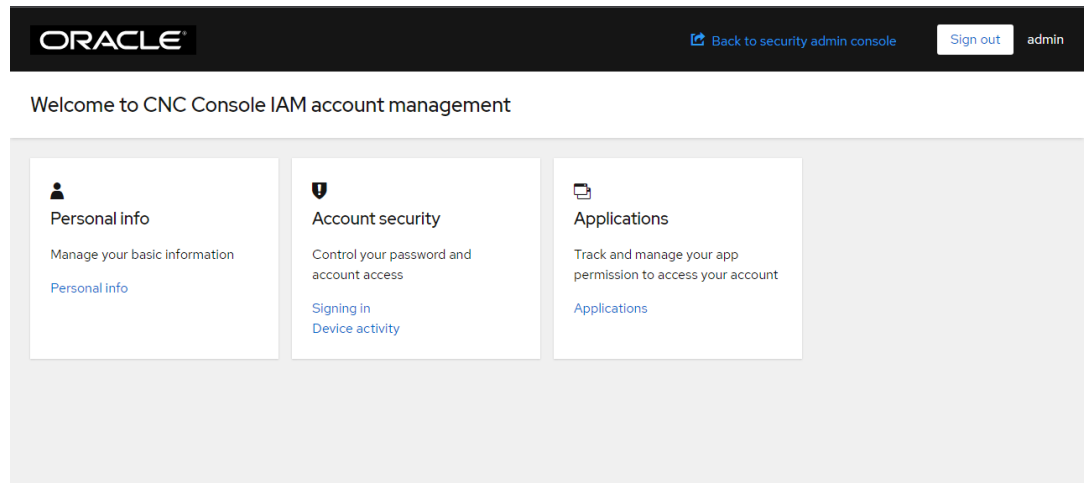


2. Click **Manage account** from the admin list.

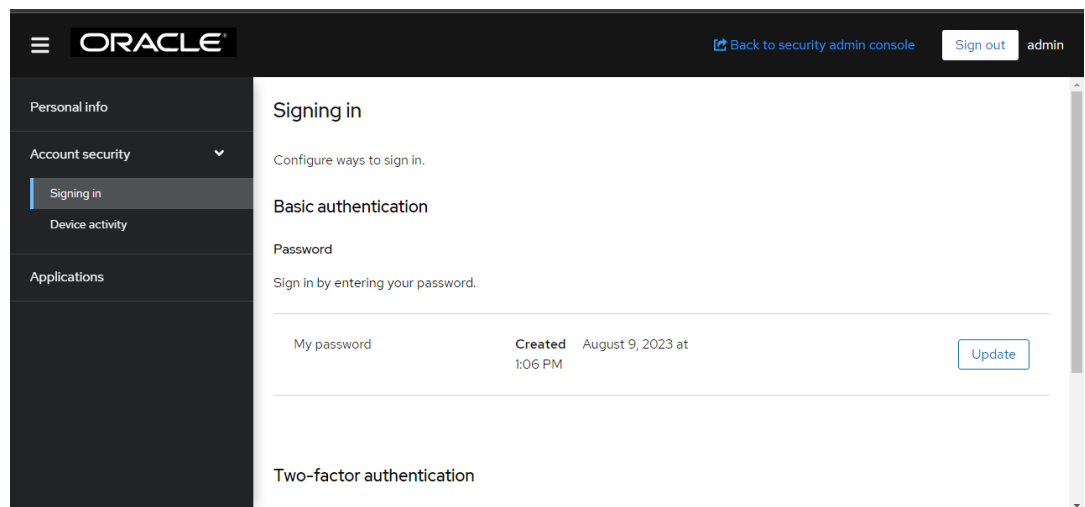
Figure 4-5 Manage Account



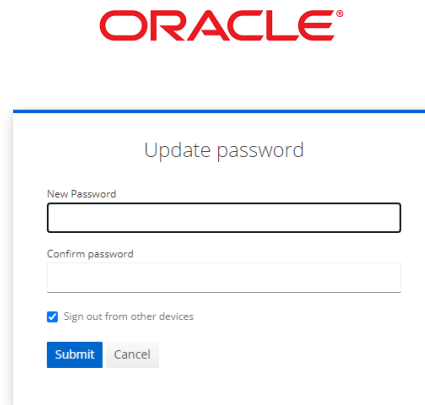
3. Click the **Account security** card.

Figure 4-6 Account Security

4. Click **Update**.

Figure 4-7 Update Password

5. Update your password on the update password screen, and click **Submit**.

Figure 4-8 Update Password Screen

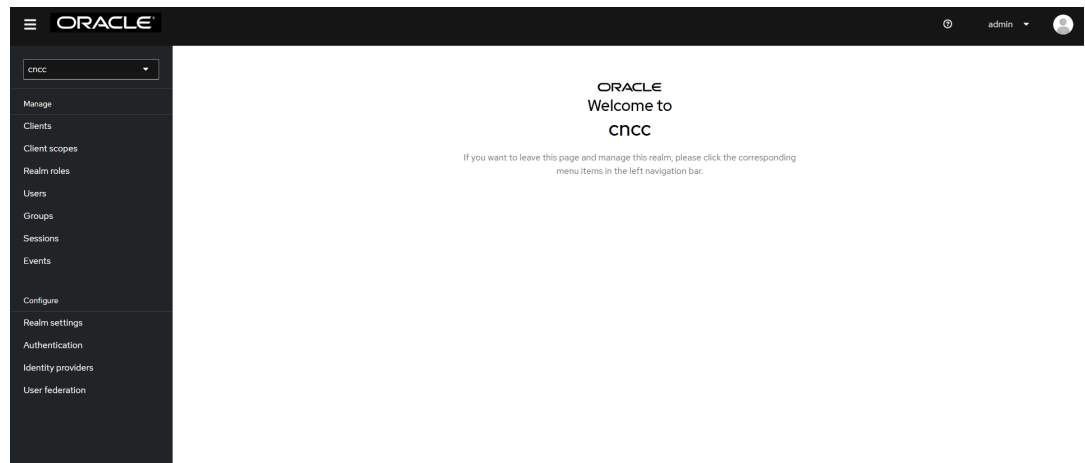
The image shows the Oracle 'Update password' screen. At the top is the Oracle logo. Below it, the title 'Update password' is centered. There are two input fields: 'New Password' and 'Confirm password'. Below these fields is a checkbox labeled 'Sign out from other devices' which is checked. At the bottom are two buttons: 'Submit' (in blue) and 'Cancel' (in grey).

4.1.4 Creating or Updating User Password in CNC Console IAM

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

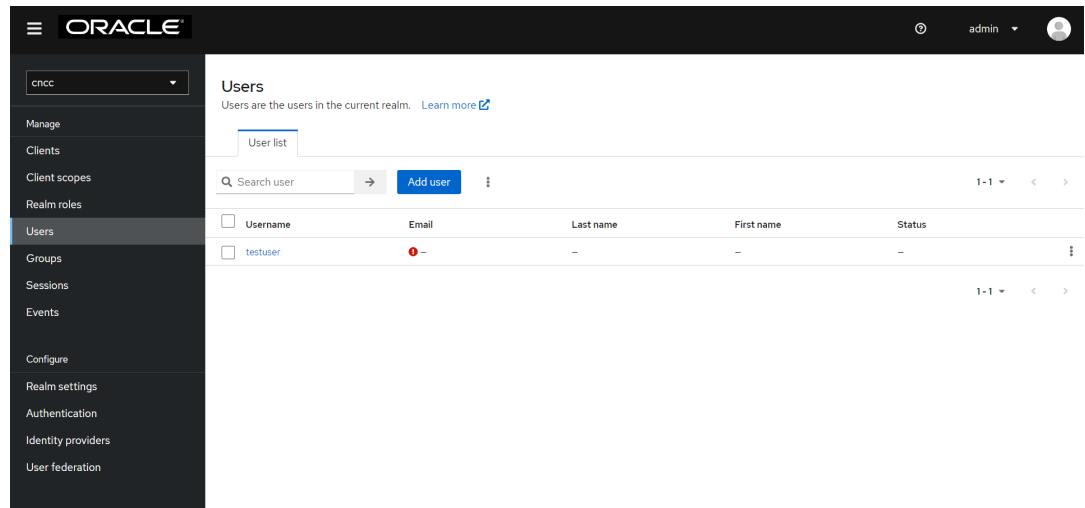
Perform the following steps to create or update the user password:

1. Select the **cncc** Realm

Figure 4-9 Realm Settings

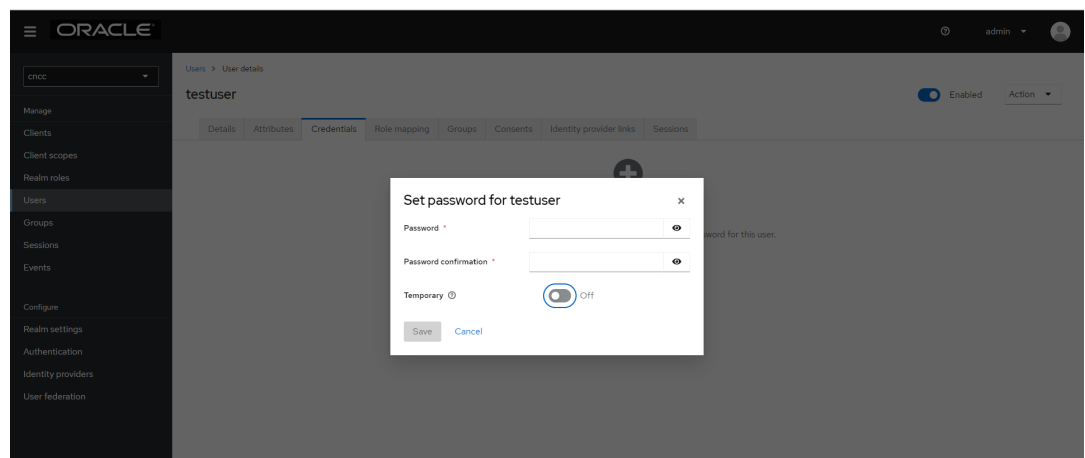
2. Click **Users** on the left pane to view all users.

Figure 4-10 Users



3. Click the **Username** of the User to update the credentials.
The following screen appears.
4. Under **Credentials** tab, click **Set Password**, set **Temporary** to **OFF**, and update the **Password**.

Figure 4-11 Credentials



4.1.5 Password Policies for CNC Console Users

The following password policies have been enabled for all CNC Console users.

Policy	Description	Value
Expire Password	The number of days the password is valid before a new password is required.	30
Special Characters	The minimum number of special characters required in the password string.	1

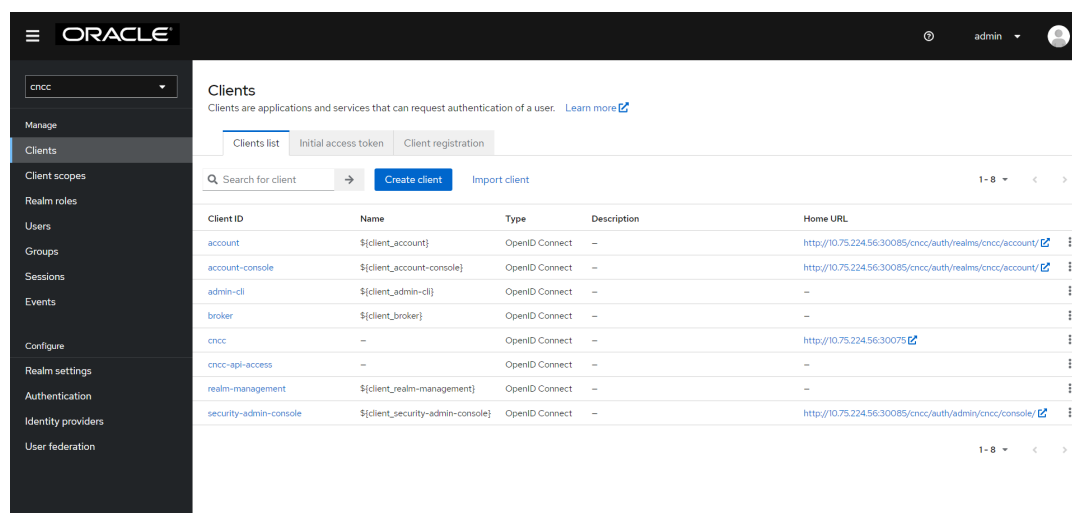
Policy	Description	Value
Uppercase Characters	The minimum number of uppercase letters required in the password string.	1
Lowercase Characters	The minimum number of lowercase letters required in the password string.	1
Digits	The minimum number of numerical digits required in the password string.	1
Not Recently Used	Prevents a recently used password from being reused.	5
Not Username	The password cannot match the username.	ON

4.2 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:

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

Figure 4-12 Clients Screen



3. Enter CNCC Core Ingress URI in the **Root URL** field and **Save**.

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

Note

Valid Redirect URIs is prepopulated, only root url needs to be configured as part of post Installation procedure.

Figure 4-13 Redirection URL

cncc - OpenID Connect
Clients are applications and services that can request authentication of a user.

Settings Roles Client scopes Sessions Advanced

General Settings

Client ID

Name

Description

Always display in UI ☐ Off

Access settings

Root URL

Home URL

Valid redirect URIs

[Add valid redirect URIs](#)

Save Revert

Jump to section

- General Settings
- Access settings
- Capability config
- Login settings
- Logout settings

4.3 Users in CNC Console IAM

This section includes:

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

Note

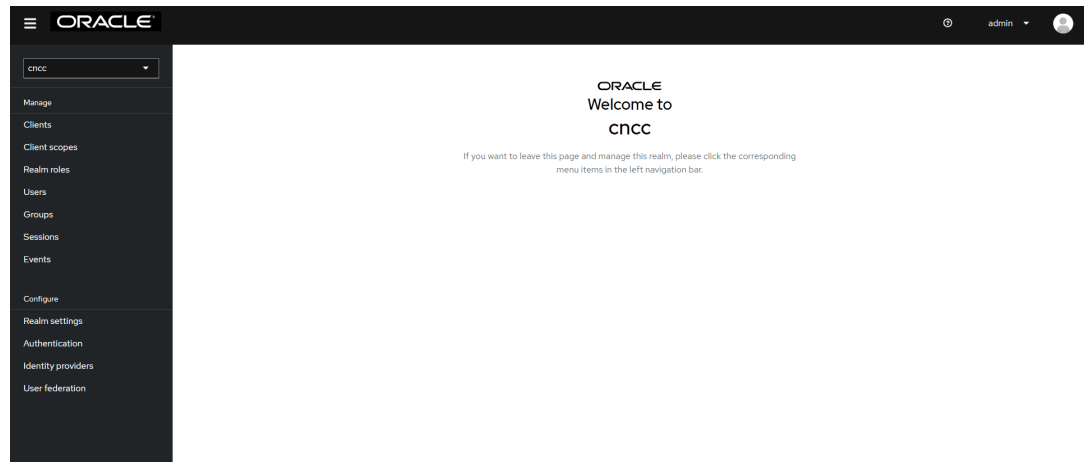
For the details about Setting or updating the admin password refer [Updating Admin Password in CNC Console IAM](#)

Note

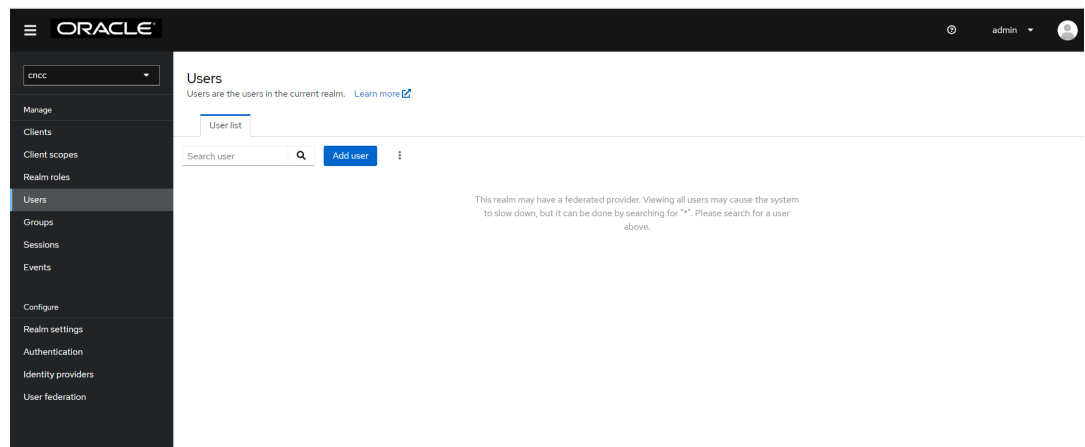
For the details about Setting or updating the user password refer [Creating or Updating User Password in CNC Console IAM](#)

4.3.1 Creating the Users

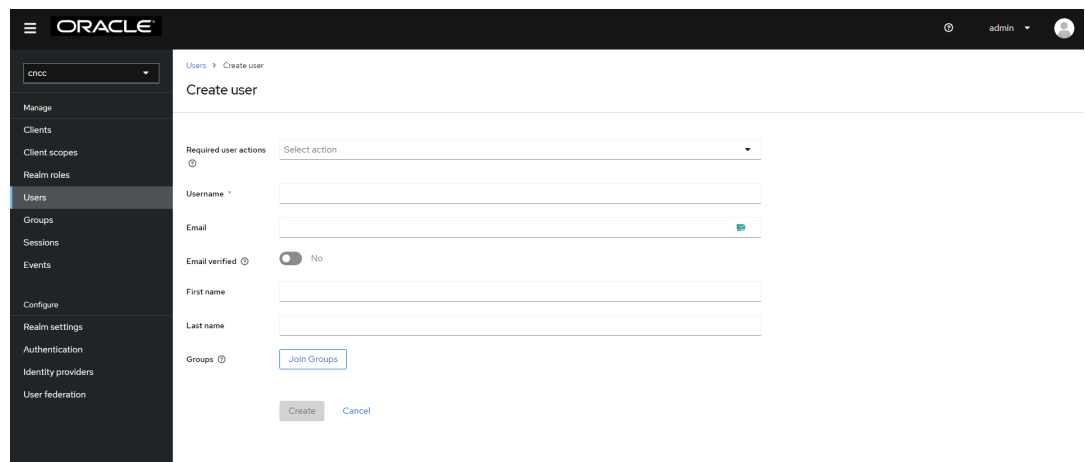
1. Select the **CNCC** Realm

Figure 4-14 CNCC Realm

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

Figure 4-15 Add User

- The **Add user** screen appears. Add the user details and click **Create**.

Figure 4-16 User Details

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

Figure 4-17 New User Created

The screenshot shows the Oracle Cloud IAM console interface. On the left is a navigation menu with options like Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main area displays the 'testuser' details page. The 'Details' tab is selected, showing fields for ID (fb79e46b-f5c0-4f02-9af4-20bf9ad3c28b), Created at (4/6/2023, 9:25:29 AM), Username (testuser), Email, and various flags like Email verified (No) and Temporarily locked (Off). There are 'Save' and 'Revert' buttons at the bottom.

- Go to the **Credentials** tab and click on **Set Password** and set the password for that user. Enable the **Temporary** flag to prompt the user to change their password when they login for the first time to CNCC GUI.

Note

You are recommended to enable the Temporary flag for security.

Figure 4-18 Set Password

The screenshot shows the Oracle Cloud IAM console interface with the 'testuser' details page. The 'Credentials' tab is selected. A 'Set password for testuser' dialog box is open, prompting for 'Password' and 'Password confirmation'. The 'Temporary' checkbox is checked, and there are 'Save' and 'Cancel' buttons at the bottom of the dialog.

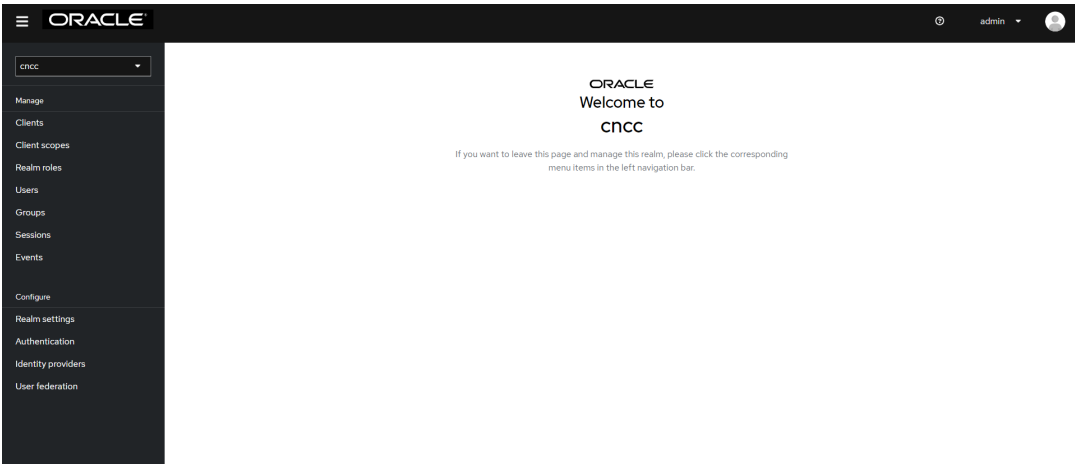
Note

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

4.3.2 Viewing the Users

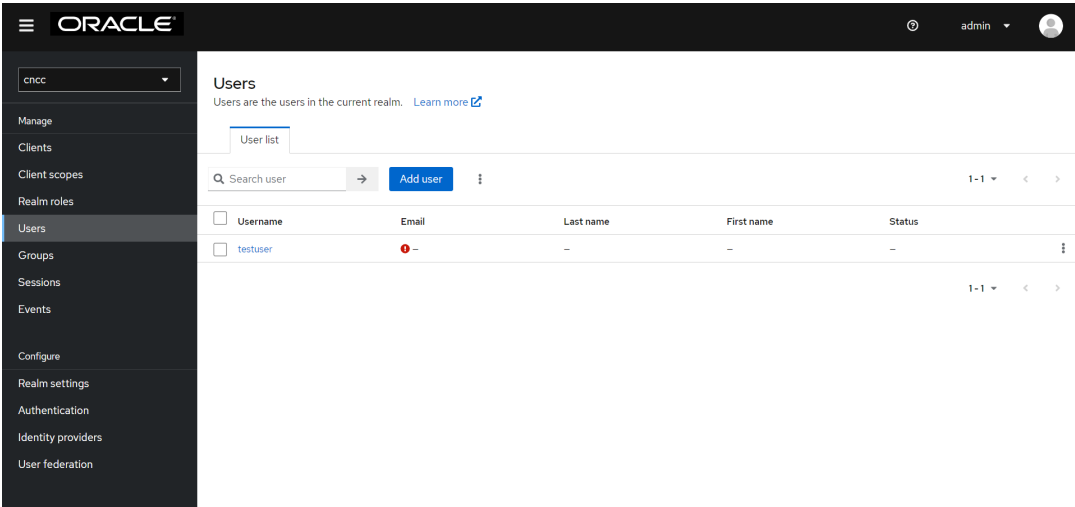
1. Select the **CNCC** Realm.

Figure 4-19 CNCC Realm



2. Select **Users** under **Manage** in the left pane to view all users.

Figure 4-20 View All Users

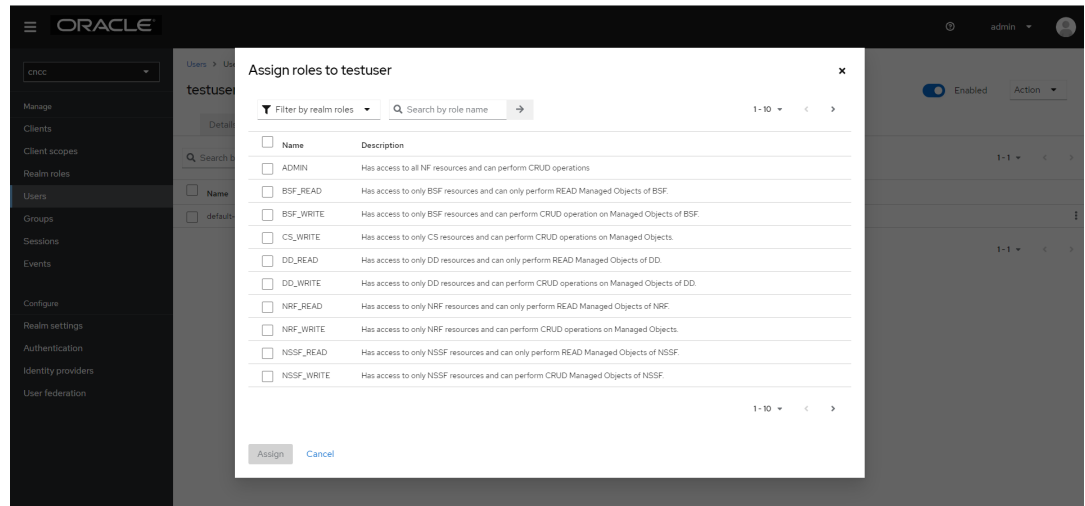


The list of users and their details appears in the right pane.

4.3.3 Assigning Roles to the User

1. Select a user. Navigate to the **Role Mappings** tab and click **Assign Role** to assign the user role.

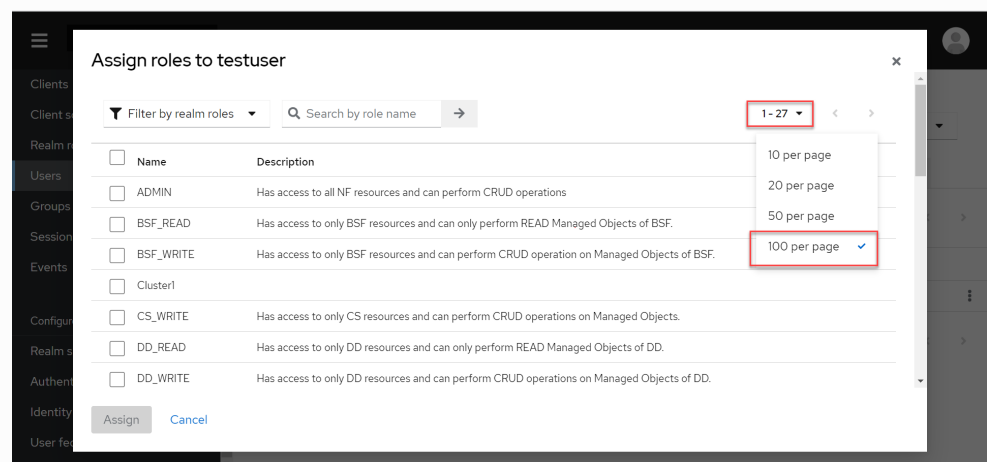
Figure 4-21 Assign Roles



Note

You must change number of entries displayed per page from the pagination dropdown to **100 per page** to view all the entries.

Figure 4-22 Display 100 Entries per Page



The selected roles will be assigned to the user.

4.4 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.

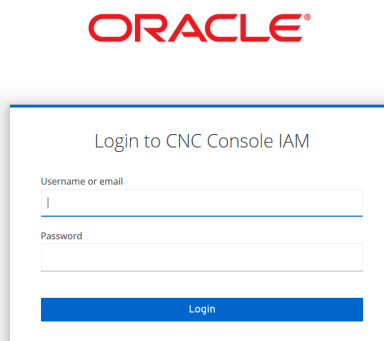
Configuring SAML Identity Provider in CNCC IAM

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-extrenal-ip>:<cncc-iam-ingress-service-port>`

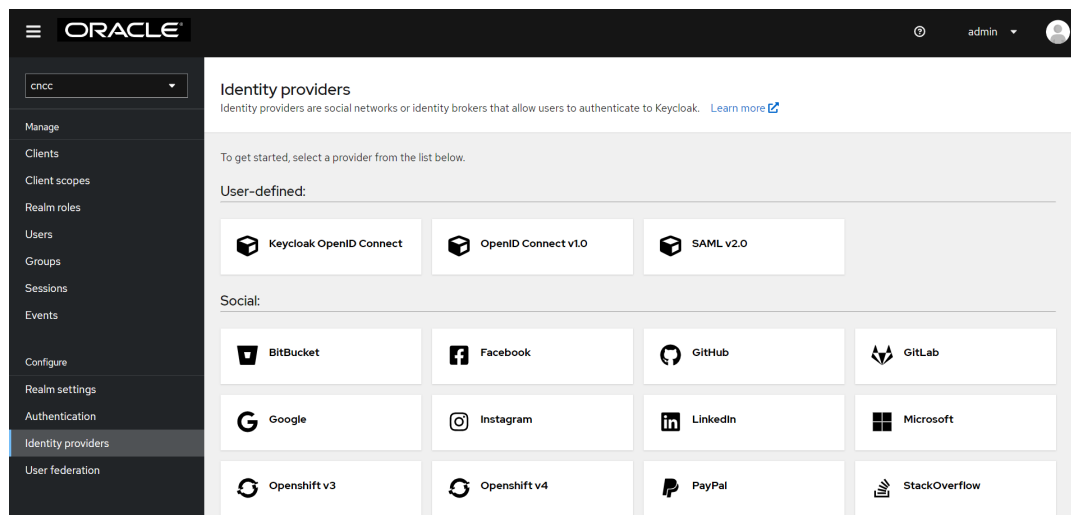
Example: `http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/`

Figure 4-23 Login screen



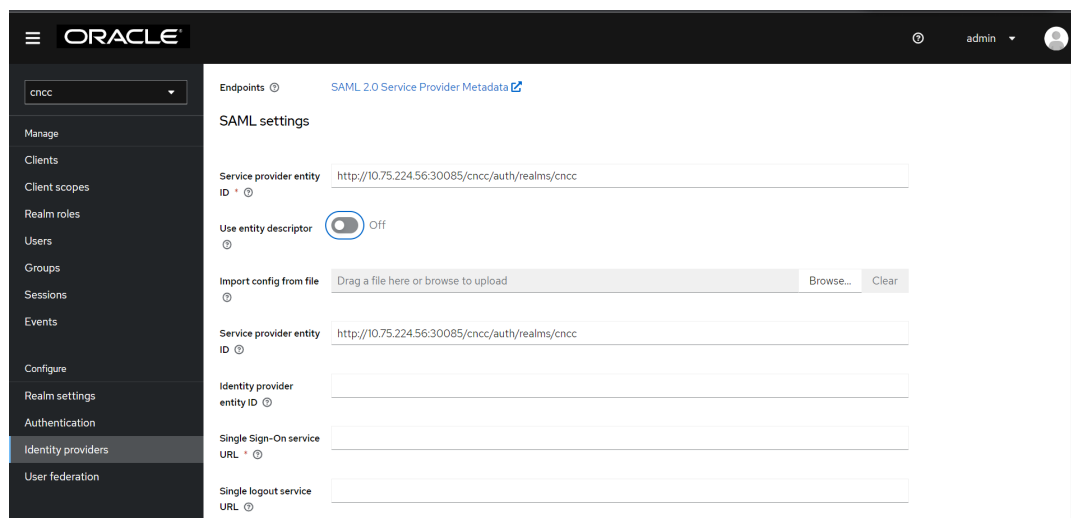
2. Click **Cncc** realm and click the **Identity providers** tab on the left pane. **Identity providers** screen appears on the right pane.

Figure 4-24 Identity Provider Screen



3. Click the **SAML v2.0** button under **User-defined**. The **Add SAML Provider** screen appears.

Figure 4-25 SAML Settings



Note

- Give an appropriate name for the field **Display Name**.
- To import the metadata file exported from SAML client in the IdP, disable the **Use Entity descriptor** flag, and upload the file from the Browse button of "Import from config file"

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

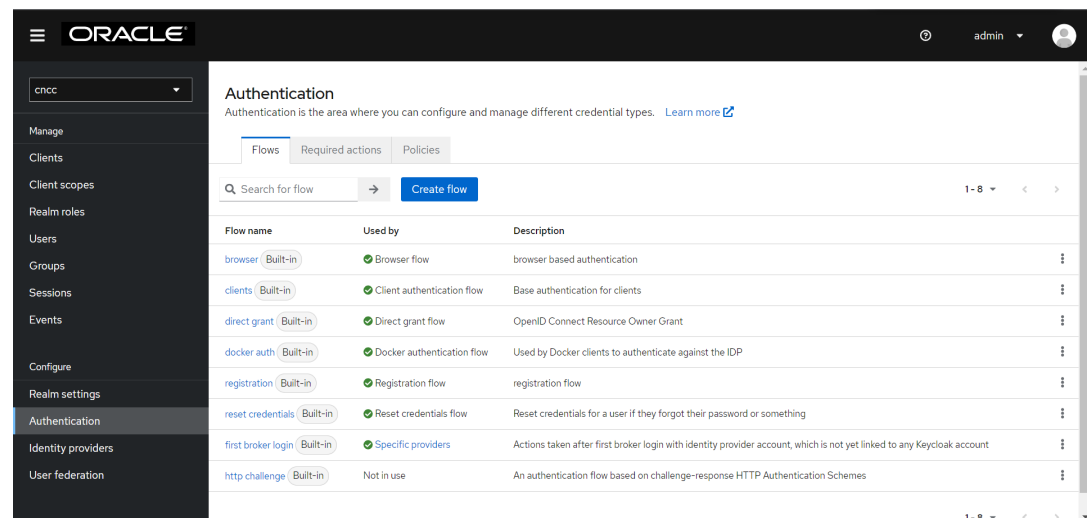
Perform the following procedure to configure the IDP manually, if you are facing difficulty in importing the metadata file from the IDP Client:

- a. Navigate to the **Identity providers** screen and click **SAML v2.0**.
- b. Set the value of **Single Sign-On Service URL** to the URL of the preferred IDP.
- c. Example: <IP/FQDN>:<PORT>/auth/realms/master/protocol/saml (URI for their preferred IDP where SAML AuthnRequest will be sent).
- d. 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).
- e. 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)
- f. If the IDP is sending signed Assertions, set **Want Assertions Signed** to **ON**.
- g. Set **Validate Signature** to **ON**.
- h. Provide value for **Validating X509 Certificates** (If you are using Keycloak as an IDP, use the certificate from **master** realm -> Realm Settings -> Keys).
- i. Click **Add**.

IDP is now configured manually.

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

Figure 4-26 Authentication



5. Click **Create Flow** on the right pane. **Create Flow** screen appears.

Figure 4-27 Create Flow

The screenshot shows the Oracle Cloud Native Configuration Console interface. On the left is a dark sidebar with a menu including 'Manage', 'Clients', 'Client scopes', 'Realm roles', 'Users', 'Groups', 'Sessions', 'Events', 'Configure', 'Realm settings', 'Authentication' (highlighted), 'Identity providers', and 'User federation'. The main content area is titled 'Authentication > Create flow'. It contains a 'Create flow' section with the text 'You can create a top level flow within this from'. Below this are three input fields: 'Name' (containing 'Simple Login Flow'), 'Description', and 'Flow type' (set to 'Basic flow'). At the bottom of this section are 'Create' and 'Cancel' buttons.

Enter the appropriate name and click **Create**.

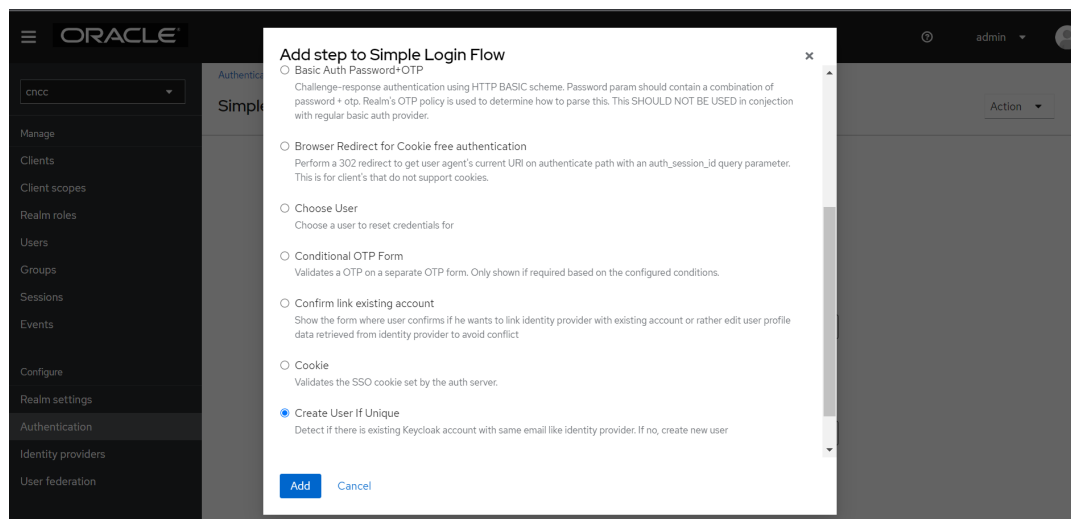
6. The **Simple Login Flow** screen appears. Click **Add execution** on the right pane.

Figure 4-28 Simple Login Flow

The screenshot shows the 'Simple Login Flow' details page in the Oracle Cloud Native Configuration Console. A notification banner at the top right says 'Flow created'. The page title is 'Simple Login Flow' with a 'Not in use' status. Below the title is a large plus icon and the text 'No steps'. A message states: 'You can start defining this flow by adding a sub-flow or an execution'. There are two sections: 'Add an execution' (describing actions like sending a reset email) with an 'Add execution' button, and 'Add a sub-flow' (describing sub-flows for user construction) with an 'Add sub-flow' button. The left sidebar is the same as in Figure 4-27.

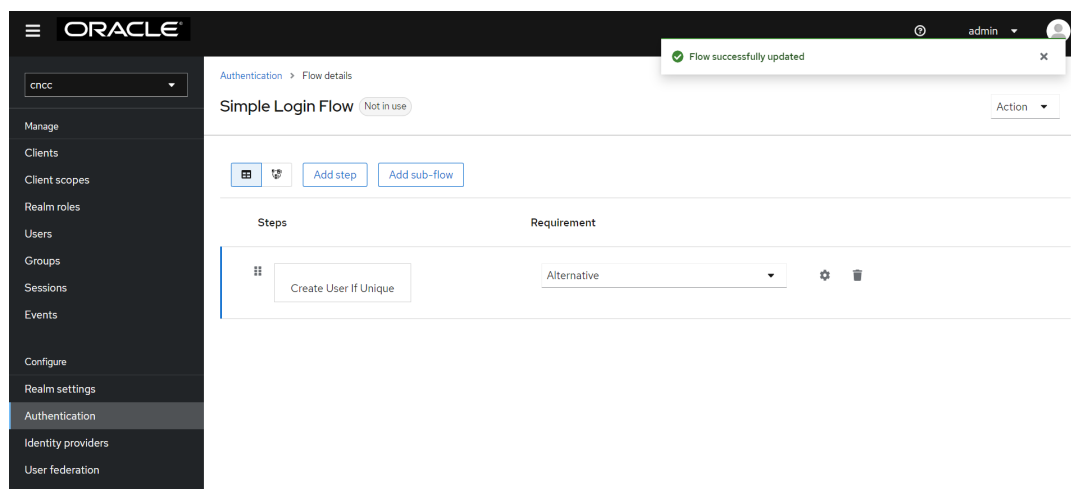
7. Select **Create User If Unique** and click **Add**.

Figure 4-29 Add Step to Simple Login Flow



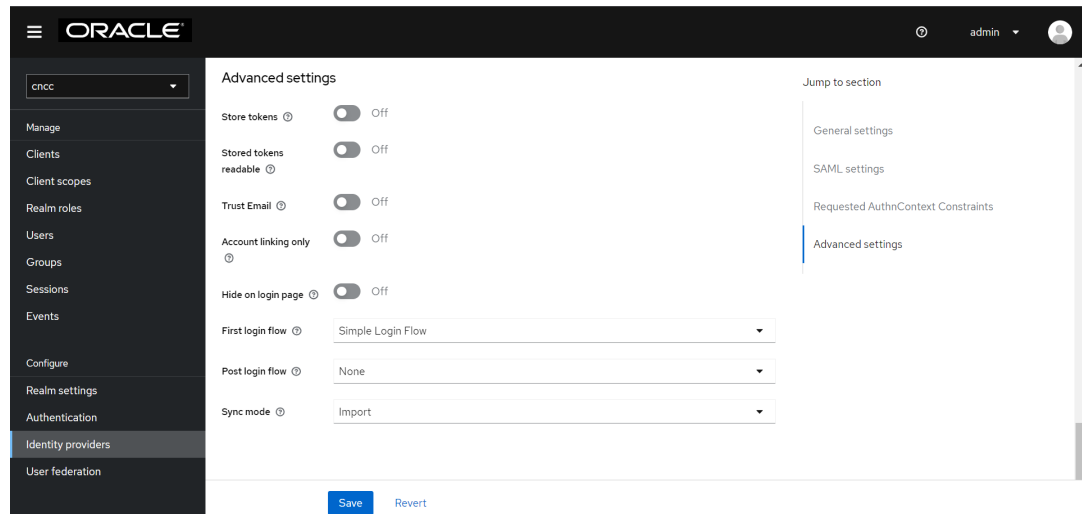
8. You will be redirected to **Authentication** page. From **Requirement** section, select **Alternative**.

Figure 4-30 Authentication



9. Click **Identity providers** in the left pane. Click the name of the Identity provide created in the previous steps, and scroll down to **Advanced Settings**. Select the custom flow from **First Login Flow** drop-down list.

Figure 4-31 Advanced Settings



10. Click **Save**.

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`

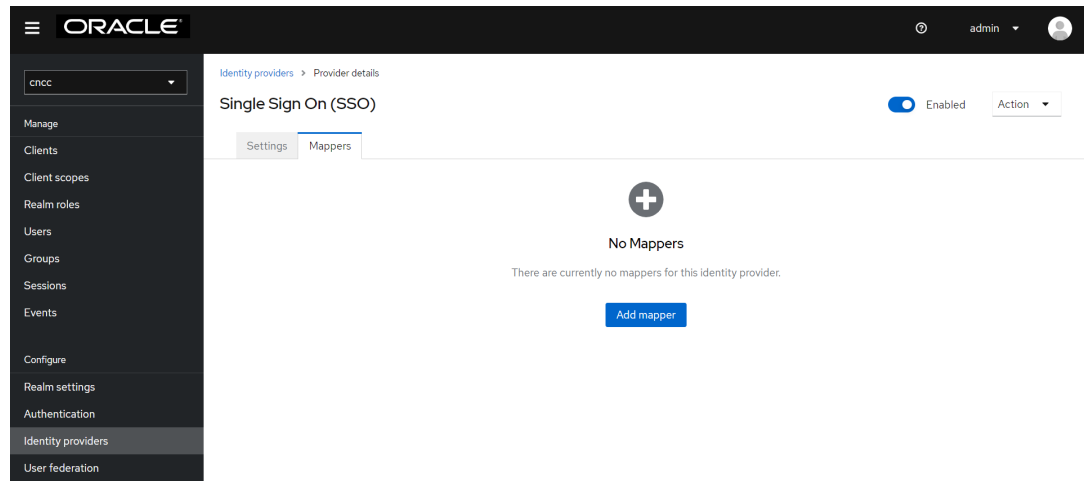
Example:

`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

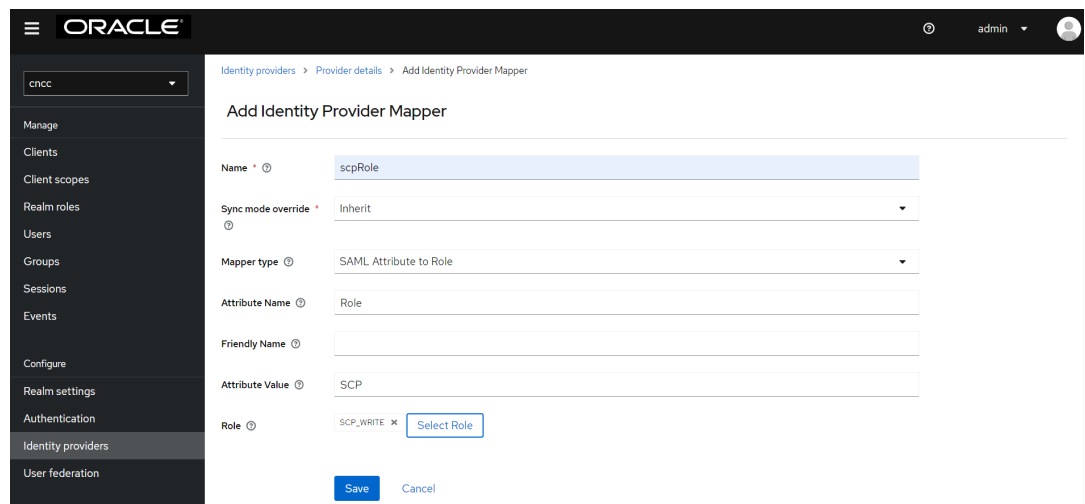
1. After saving SAML IdP configurations in CNCC IAM, select **Identity providers** on the left pane and click the name of your identity provider. Click **Mappers** tab on the right pane. Click **Add Mapper**.

Figure 4-32 Single Sign On

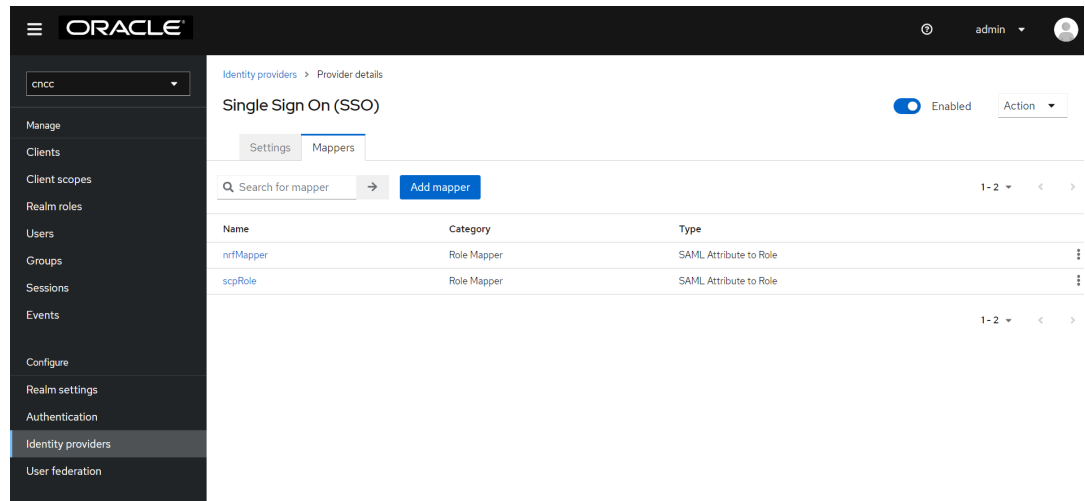


2. Add Identity Provider Mapper Screen appears.

Figure 4-33 Add Identity Provider Mapper



- Give an appropriate name for the Identity Provider Mapper in the **Name** field.
- 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 **Assign**. Then click **Save**

Figure 4-34 Single Sign On

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/`



Login to CNCC

Username or email

Password

Or sign in with

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.

4.5 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

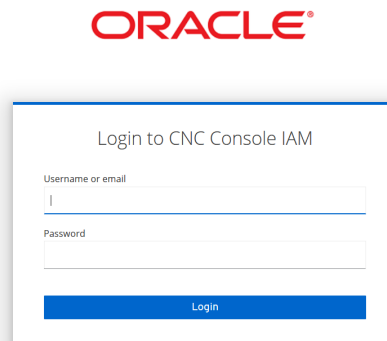
dn: cn=nrf,ou=groups,dc=oracle,dc=org
objectclass: top
objectclass: groupOfUniqueNames
cn: nrf
uniqueMember: uid=ben,ou=people,dc=oracle,dc=org
uniqueMember: uid=bob,ou=people,dc=oracle,dc=org
ou: nrfusers
```

4.5.1 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.

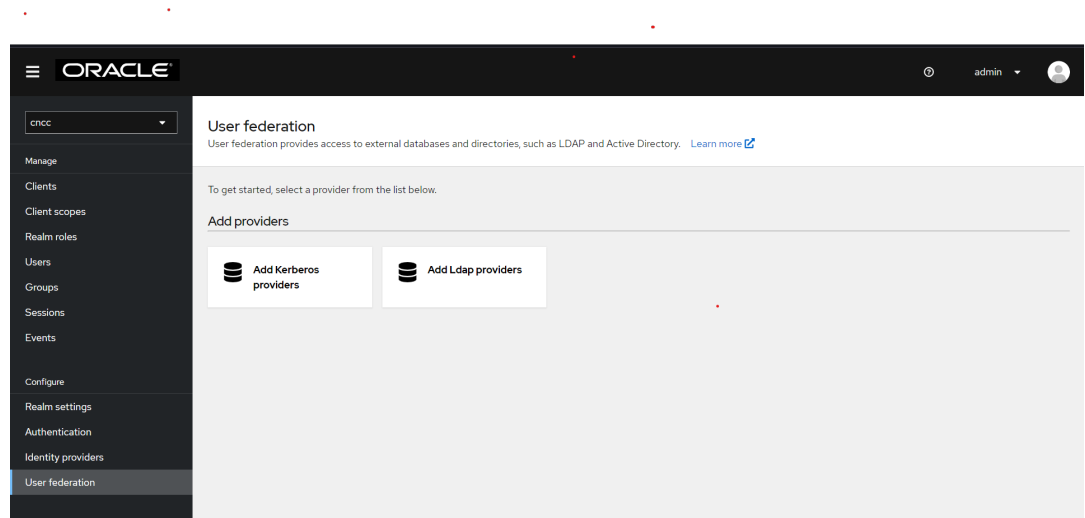
Figure 4-35 Login Screen



The login screen features the Oracle logo at the top. Below it, the title "Login to CNC Console IAM" is centered. There are two input fields: "Username or email" and "Password". A blue "Login" button is positioned at the bottom of the form.

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 4-36 User Federation



3. Click **Add LDAP providers**. The following page will automatically open a form to fill in you LDAP connection parameters. The form will be initially empty as shown below:

Figure 4-37 Add LDAP providers

4. Enter the values for the following fields:
- **UI Display Name:** Enter the display name.
 - **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.

- provide your company LDAP server details in the **Connection URL** field, same as you provided for *ldap-ldif* file already i.e. connection url (hostname prefixed with ldap:// (OR) when LDAP Secure connection enabled(LDAPS) hostname prefix should be ldaps://), the port.

Figure 4-38 General Options

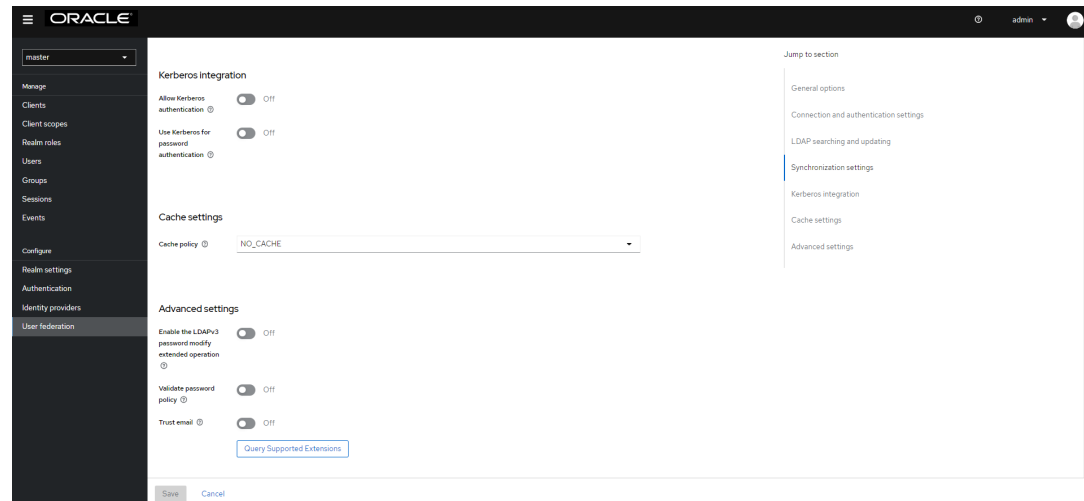
- If your LDAP is secured then select **simple** from the **Bind Type** drop down and add the admin bind username and password or select Bind-type as **none**. Sample data for the field Bind DN: **"cn=admin,dc=oracle,dc=org"**
- Click **Test Connection** and **Test Authentication**. Both these tests will be successful.
- Proceed to the **Edit Mode** drop down list. Select **READ_ONLY**
- In most cases the **UUID LDAP attribute** value set as entry UUID. If you do not have a suitable value, use an alternate unique identifier.
- Click **Test Connection** and **Test Authentication**.

Figure 4-39 User Federation

- The default setting for **Import Users** is **ON**. Change it to **OFF** to disable user sync.
 - Set Cache policy as **NO_CACHE**
5. After filling the required fields, the screen appears as below.

Figure 4-40 User Federation

6. Click **Save**.

Figure 4-41 User Federation

4.5.2 Grouping the LDAP Mapper and Assigning the Roles

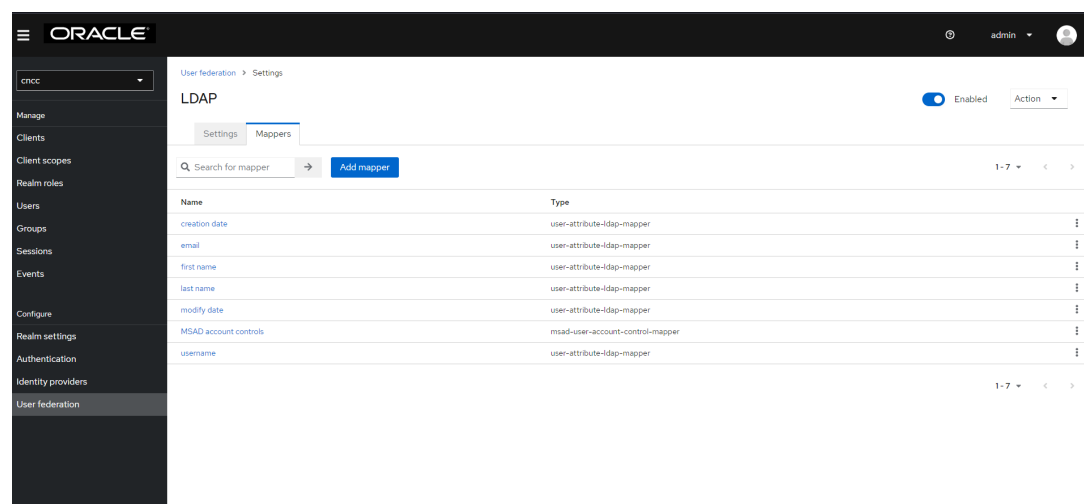
When an LDAP Federation provider is created, CNC Console-IAM provides a set of built-in mappers for this provider. User can change the set and create a new mapper or update and delete existing ones.

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. Under **Configure** in the left pane, click **User Federation**. Click **Idap** and select the **Mappers** tab, and click **Add Mapper**.

Figure 4-42 LDAP Mapper Page

2. The **Create New Mapper** page appears. Give an appropriate name for the field **Name**. Select group-ldap-mapper as **Mapper Type** drop down menu. Click **Save**.

Figure 4-43 User Federation Mapper Page

The screenshot shows the Oracle Cloud Native Configuration Console interface. On the left is a dark sidebar with a menu. The top of the sidebar has the Oracle logo and a hamburger menu icon. Below it is a dropdown menu currently showing 'cncc'. The menu items are: Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation (which is highlighted). The main content area has a breadcrumb trail: 'User federation > Settings > Mapper details'. Below the breadcrumb is the title 'Create new mapper'. The form contains two input fields: 'Name' and 'Mapper type', both with a question mark icon to their left. At the bottom of the form are two buttons: 'Save' (in blue) and 'Cancel'.

The following screen appears.

Figure 4-44 LDAP Mapper Filled Form

This screenshot shows the same 'Create new mapper' page, but now it is filled with configuration data. The 'Name' field contains 'group-mapper' and the 'Mapper type' dropdown is set to 'group-ldap-mapper'. The 'LDAP Groups DN' field contains 'ou=groups,dc=oracle,dc=org'. The 'Group Name LDAP Attribute' field contains 'cn'. The 'Group Object Classes' field contains 'groupOfUniqueNames, top'. There are two toggle switches: 'Preserve Group Inheritance' and 'Ignore Missing Groups', both currently set to 'Off'. The 'Membership LDAP Attribute' field contains 'uniqueMember'. The 'Membership Attribute Type' dropdown is set to 'DN'. The 'Membership User LDAP Attribute' field contains 'cn'. The 'LDAP Filter' field is empty. The 'Mode' dropdown is set to 'READ_ONLY'. The 'User Groups Retrieve Strategy' dropdown is set to 'LOAD_GROUPS_BY_MEMBER_ATTRIBUTE'. The 'Member-Of LDAP Attribute' field contains 'memberOf'.

Note

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

3. Click **Save**.

Figure 4-45 Save

Oracle Cloud Native Configuration Console - User federation settings page. The left sidebar shows the navigation menu with 'User federation' selected. The main content area displays the following configuration:

- User Groups Retrieve Strategy: LOAD_GROUPS_BY_MEMBER_ATTRIBUTE
- Member-Of LDAP Attribute: memberOf
- Mapped Group Attributes: (empty)
- Drop non-existing groups during sync: Off
- Groups Path: /

Buttons: Save, Cancel

- click the name of your mapper. Under Action menu, click **Sync LDAP Groups to Keycloak**. The success message appears with the number of groups imported and so on.

Figure 4-46 Group Mapper

Oracle Cloud Native Configuration Console - group-mapper details page. The left sidebar shows the navigation menu with 'User federation' selected. The main content area displays the following configuration:

- ID: 790b423e-2394-4853-b69f-82fab602ba4
- Name: group-mapper
- Mapper type: group-ldap-mapper
- LDAP Groups DN: ou=groups,dc=oracle,dc=org
- Group Name LDAP Attribute: cn
- Group Object Classes: groupOfUniqueNames, top
- Preserve Group Inheritance: Off
- Ignore Missing Groups: Off
- Membership LDAP Attribute: uniqueMember
- Membership Attribute Type: DN
- Membership User LDAP Attribute: cn
- LDAP Filter: (empty)

Buttons: Action

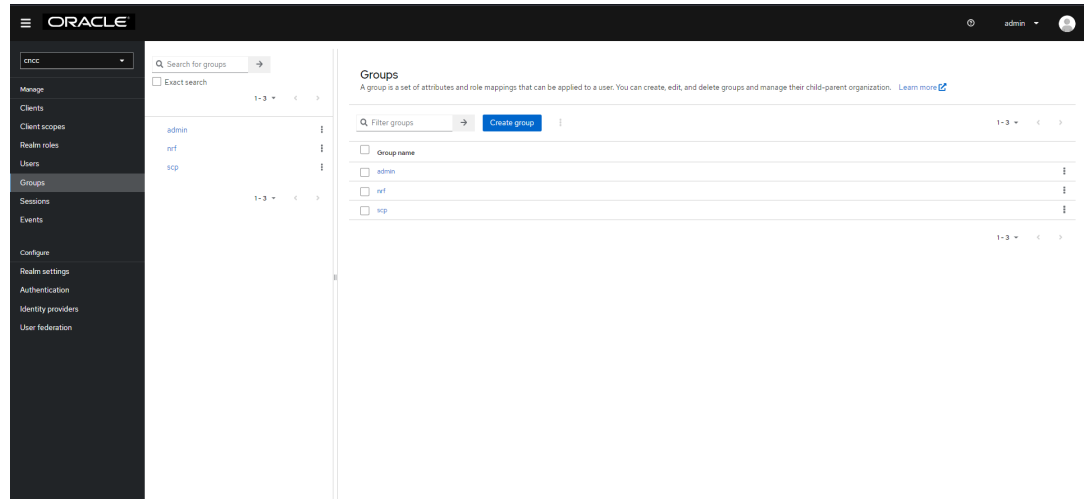
Success message: Data successfully synced: 0 imported groups, 3 updated groups, 0 removed groups

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 Configuration Console Troubleshooting Guide* and see [CNC Console Logs](#) for further details.

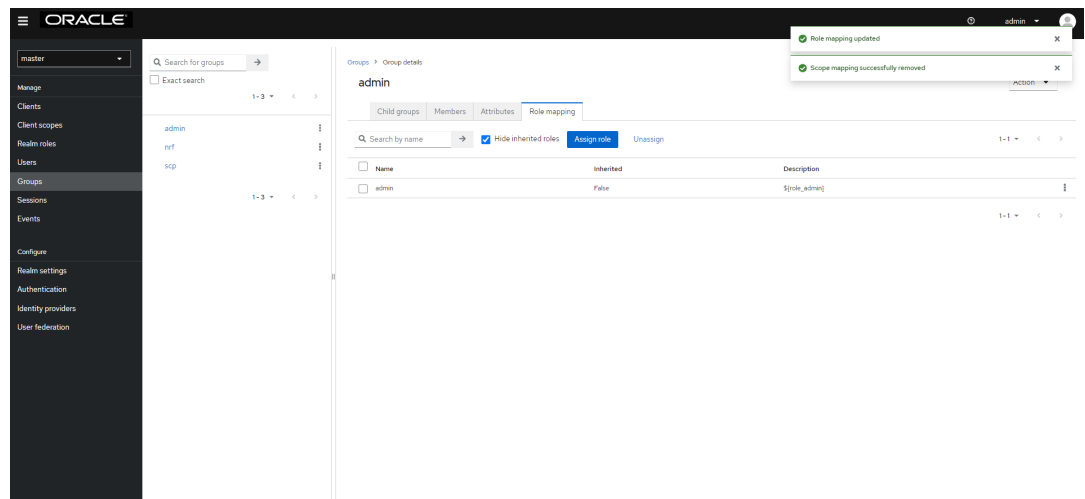
- Select the **Groups** in the left pane to view all groups.

Figure 4-47 Groups



- Click any group and click **Edit**. The following tabs appear: **Child groups**, **Attributes**, **Role Mappings**, and **Members**.

Figure 4-48 Role mapping to LDAP Group



- 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

Accessing NF Configurations through 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.

5.1 Generate Access Tokens

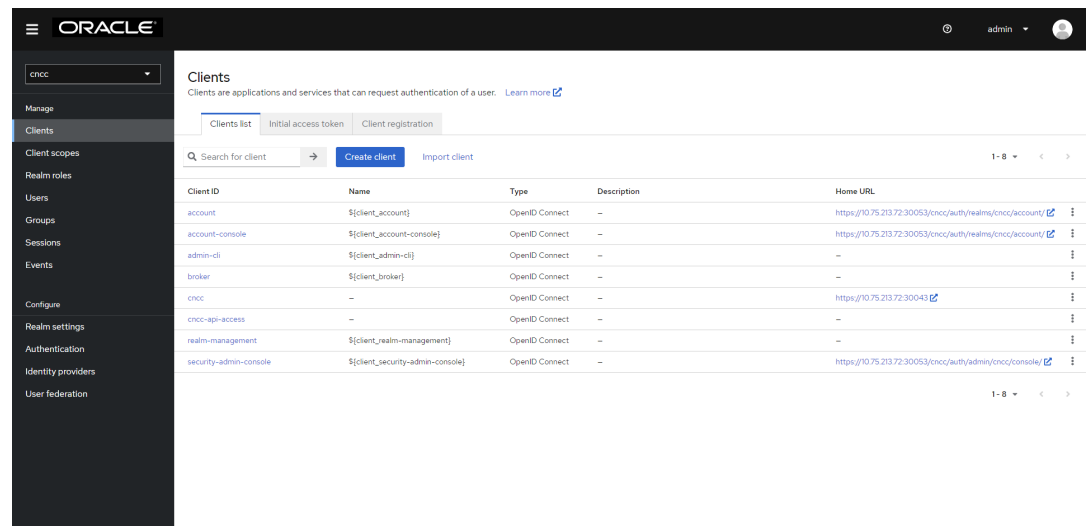
CNCC IAM provides a REST API for generating and refreshing access tokens.

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 the **cncc** realm.
3. On the right pane, click **Clients**. The following screen appears:

Figure 5-1 Clients



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

Figure 5-2 Cncc-api-access

5. Navigate to the Capability Config section in the Settings tab and select the **Direct Access Grants** checkbox. Click **Save**.

Figure 5-3 Direct Access Grants

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://${cncc-iam-ingress-external-ip}:${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-urlencoded* 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=<password value>' \
--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":
  "eyJhbGciOiJSUzI1NiIsI....._LcCZYwDQJJTloj2PJ8y1Wj09l2Q",
  "expires_in": 300,
  "refresh_expires_in": 1800,
  "refresh_token":
  "eyJhbGciOiJIUzI1NiIsI.....ldUIiwia2lkIiA6ICI3YTFlYvKPF-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.

5.2 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:

```
'client_id': 'cncc-api-access',
'refresh_token':
'eyJhbGciOiJIUzI1NiIs ....ldUIiwia2lkIiA6ICI3YTFlYVVKPF-ZIg',
'grant_type': 'refresh_token'
```

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

5.3 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

CNC Console Metrics

This section provides the information about CNC Console Metrics.

Note

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 (OCCNE 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 (OCCNE 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

Dimension Description

The following table describes the different types of metric dimensions:

Table 6-1 CNC Console Dimension Descriptions

Dimension	Description	Values
Method	Http method	GET, PUT, POST, DELETE, PATCH

Table 6-1 (Cont.) CNC Console Dimension Descriptions

Dimension	Description	Values
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
Host		
Status		
ErrorOriginator		

6.1 CNC Console IAM Metrics

This section provides the information about the CNCC IAM Metrics:

6.1.1 M-CNCC IAM Requests

Table 6-2 M-CNCC IAM Requests

Metric Details	Total number of requests received by M-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

6.1.2 M-CNCC IAM Response

Table 6-3 M-CNCC IAM Response

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

Table 6-3 (Cont.) M-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

6.1.3 M-CNCC IAM Success Responses

Table 6-4 M-CNCC IAM Success Responses

Metric Details	Total number of success responses (2xx) for M-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

6.1.4 M-CNCC IAM 5xx Responses

Table 6-5 M-CNCC IAM 5xx Responses

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

Table 6-5 (Cont.) M-CNCC IAM 5xx Responses

Dimensions	- ErrorOriginator - Host - HttpVersion - InstanceIdIdentifier - Method - Route_path - Scheme - Status - ResourcePath - ResourceType - UserId - UserName
Metric Type	Counter

6.1.5 M-CNCC IAM 4xx Responses

Table 6-6 M-CNCC IAM 4xx Responses

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

6.1.6 M-CNCC IAM Error Responses

Table 6-7 M-CNCC IAM Error Responses

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

Table 6-7 (Cont.) M-CNCC IAM Error Responses

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

6.1.7 M-CNCC IAM Access Token Request

Table 6-8 M-CNCC IAM Access Token Request

Metric Details	Total number of access token requests received for M-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

6.1.8 M-CNCC IAM Access Token Granted

Table 6-9 M-CNCC IAM Access Token Granted

Metric Details	Total number of access token granted for M-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

6.1.9 M-CNCC IAM Access Token Not Granted

Table 6-10 M-CNCC IAM Access Token Not Granted

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

Table 6-10 (Cont.) M-CNCC IAM Access Token Not Granted

Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token",Status=~"4.* 5.*"} <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=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token",Status=~"4.* 5.*"}) <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=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/master/protocol/openid-connect/token",Status=~"4.* 5.*"})
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName
Metric Type	Counter

6.1.10 M-CNCC IAM User Login Failure Responses

Table 6-11 M-CNCC IAM User Login Failure Responses

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

Table 6-11 (Cont.) M-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

6.2 M-CNCC Core Metrics

This section provides the information about the M-CNCC Core Metrics:

6.2.1 M-CNCC Core Requests

Table 6-12 M-CNCC Core Requests

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

Table 6-12 (Cont.) M-CNCC Core Requests

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

6.2.2 M-CNCC Core Responses

Table 6-13 M-CNCC Core Responses

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

6.2.3 M-CNCC Core Success Responses

Table 6-14 M-CNCC Core Success Responses

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

Table 6-14 (Cont.) M-CNCC Core Success Responses

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

6.2.4 M-CNCC Core 5xx Responses

Table 6-15 M-CNCC Core 5xx Responses

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

6.2.5 M-CNCC Core 4xx Responses

Table 6-16 M-CNCC Core 4xx Responses

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

Table 6-16 (Cont.) M-CNCC Core 4xx Responses

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

6.2.6 M-CNCC Core Error Responses

Table 6-17 M-CNCC Core Error Responses

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

6.2.7 M-CNCC Core Access Token Request

Table 6-18 M-CNCC Core Access Token Request

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

Table 6-18 (Cont.) M-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 or older)</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: (OCCNE 1.9.x and higher 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

6.2.8 M-CNCC Core Access Token Granted Responses

Table 6-19 M-CNCC Core Access Token Granted Responses

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

Table 6-19 (Cont.) M-CNCC Core Access Token Granted Responses

Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status="200 OK"} <i>Group by user: (CNE 1.8 or older)</i> 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="200 OK"}) <i>Group by user: (OCCNE 1.9.x and higher versions)</i> 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="200 OK"})
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

6.2.9 M-CNCC Core Access Token Not Granted Responses

Table 6-20 M-CNCC Core Access Token Granted Responses

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

Table 6-20 (Cont.) M-CNCC Core Access Token Granted Responses

Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status=~"4.* 5.*"} <i>Group by user: (CNE 1.8 or older)</i> 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.*"}) <i>Group by user: (OCCNE 1.9.x and higher versions)</i> 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.*"})
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

6.2.10 M-CNCC Core User Login Failure Responses

Table 6-21 M-CNCC Core User Login Failure Responses

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

Table 6-21 (Cont.) M-CNCC Core User Login Failure Responses

Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/login-actions/authenticate",Status="200 OK"} <i>Group by user: (CNE 1.8 or older)</i> 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"}) <i>Group by user: (OCCNE 1.9.x and higher versions)</i> 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"})
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

6.2.11 M-CNCC Core User Authorization Failure Responses

Table 6-22 M-CNCC Core User Authorization Failure Responses

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

Table 6-22 (Cont.) M-CNCC Core User Authorization Failure Responses

Metric Filter	<code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status="403 FORBIDDEN", ResourceType!="UNKNOWN"}</code> <i>Group by user: (CNE 1.8 or older)</i> <code>sum by(Status,Method,kubernetes_namespace,ResourceType,UserId,UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status="403 FORBIDDEN", ResourceType!="UNKNOWN"})</code> <i>Group by user: (OCCNE 1.9.x and higher versions)</i> <code>sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status="403 FORBIDDEN", ResourceType!="UNKNOWN"})</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

6.2.12 M-CNCC Core BSF Requests

Table 6-23 M-CNCC Core BSF Requests

Metric Details	Total number of requests received by M-CNCC Core for BSF
Metric Filter	<code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/bsfapi/*"}</code> Multiple Route_path: <code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/bsfapi/* .*oc-bsf-configuration/* .*bsf/*"}</code>

Table 6-23 (Cont.) M-CNCC Core BSF Requests

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

6.2.13 M-CNCC Core BSF Responses

Table 6-24 M-CNCC Core BSF Responses

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

6.2.14 M-CNCC Core DD Requests

Table 6-25 M-CNCC Core DD Requests

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

Table 6-25 (Cont.) M-CNCC Core DD Requests

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

6.2.15 M-CNCC Core DD Responses

Table 6-26 M-CNCC Core DD Responses

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

6.2.16 M-CNCC Core PROVGW Requests

Table 6-27 M-CNCC Core PROVGW Requests

Metric Details	Total number of requests received by M-CNCC Core for PROVGW
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/.*prov-gw-config/.*/.*prov-gw/.*/.*"}

Table 6-27 (Cont.) M-CNCC Core PROVGW Requests

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

6.2.17 M-CNCC Core PROVGW Responses

Table 6-28 M-CNCC Core PROVGW Responses

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

6.2.18 M-CNCC Core NRF Responses

Table 6-29 M-CNCC Core NRF Responses

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

Table 6-29 (Cont.) M-CNCC Core NRF Responses

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

6.2.19 M-CNCC Core NRF Requests

Table 6-30 M-CNCC Core NRF Requests

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

6.2.20 M-CNCC Core NSSF Requests

Table 6-31 M-CNCC Core NSSF Requests

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

6.2.21 M-CNCC Core NSSF Responses

Table 6-32 M-CNCC Core NSSF Responses

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

6.2.22 M-CNCC Core NWDAF Requests

Table 6-33 M-CNCC Core NWDAF Requests

Metric Details	Total number of requests received by M-CNCC Core for NWDAF
Metric Filter	<code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ocnwdaf.*/*ocnwdafapi.*"}</code>

Table 6-33 (Cont.) M-CNCC Core NWDAF Requests

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

6.2.23 M-CNCC Core NWDAF Responses

Table 6-34 M-CNCC Core NWDAF Responses

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

6.2.24 M-CNCC Core POLICY Requests

Table 6-35 M-CNCC Core POLICY Requests

Metric Details	Total number of requests received by M-CNCC Core for POLICY
Metric Filter	Single Route_path: oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/policyapi/*"} Multiple Route_path: oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/policyapi/* .*/oc-cnpolicy-configuration/* .*/pcf/*"}

Table 6-35 (Cont.) M-CNCC Core POLICY Requests

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

6.2.25 M-CNCC Core POLICY Responses

Table 6-36 M-CNCC Core POLICY Responses

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

6.2.26 M-CNCC Core SCP Responses

Table 6-37 M-CNCC Core SCP Responses

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

Table 6-37 (Cont.) M-CNCC Core SCP Responses

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

6.2.27 M-CNCC Core SCP Requests

Table 6-38 M-CNCC Core SCP Requests

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

6.2.28 M-CNCC Core SEPP Requests

Table 6-39 M-CNCC Core SEPP Requests

Metric Details	Total number of requests received by M-CNCC Core for SEPP
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/sepp-configuration/.*"} Multiple Route_path: oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/sepp-configuration/.*/sepp/.*"}

Table 6-39 (Cont.) M-CNCC Core SEPP Requests

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

6.2.29 M-CNCC Core SEPP Responses

Table 6-40 M-CNCC Core SEPP Responses

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

6.2.30 M-CNCC Core UDR Requests

Table 6-41 M-CNCC Core UDR Requests

Metric Details	Total number of requests received by M-CNCC Core for UDR
----------------	--

Table 6-41 (Cont.) M-CNCC Core UDR Requests

Metric Filter	Single Route_path: oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/nudr-config/*"} Multiple Route_path: oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/nudr-dr-prov/* .*n5g-eir-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

6.2.31 M-CNCC Core UDR Responses

Table 6-42 M-CNCC Core UDR Responses

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

6.2.32 M-CNCC Core cnDBTier Requests

Table 6-43 M-CNCC Core cnDBTier Requests

Metric Details	Total number of requests received by M-CNCC Core for cnDBTier
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~\".*ocdbtier.*\"}
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • Username • AuthenticationType
Metric Type	Counter

6.2.33 M-CNCC Core cnDBTier Responses

Table 6-44 M-CNCC Core cnDBTier Responses

Metric Details	Total number of responses sent by M-CNCC Core for cnDBTier
Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status="200 OK",Method="GET",ResourcePath=~".*ocdbtier.*"}
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • Username • AuthenticationType
Metric Type	Counter

6.2.34 M-CNCC Core OCCM Requests

Table 6-45 M-CNCC Core OCCM Requests

Metric Details	Total number of requests received by M-CNCC Core for OCCM
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*occm-config.*"}

Table 6-45 (Cont.) M-CNCC Core OCCM Requests

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

6.2.35 M-CNCC Core OCCM Responses

Table 6-46 M-CNCC Core OCCM Responses

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

6.3 A-CNCC Core Metrics

This section provides the information about the A-CNCC Core Metrics:

6.3.1 A-CNCC Core Requests

Table 6-47 A-CNCC Core Requests

Metric Details	Total number of requests received by A-CNCC Core
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway"}

Table 6-47 (Cont.) A-CNCC Core Requests

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

6.3.2 A-CNCC Core Responses

Table 6-48 A-CNCC Core Responses

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

6.3.3 A-CNCC Core Success Responses

Table 6-49 A-CNCC Core Success Responses

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

Table 6-49 (Cont.) A-CNCC Core Success Responses

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

6.3.4 A-CNCC Core 5xx Responses

Table 6-50 A-CNCC Core 5xx Responses

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

6.3.5 A-CNCC Core 4xx Responses

Table 6-51 A-CNCC Core 4xx Responses

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

Table 6-51 (Cont.) A-CNCC Core 4xx Responses

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

6.3.6 A-CNCC Core Error Responses

Table 6-52 A-CNCC Core Error Responses

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

6.3.7 A-CNCC Core Access Token Request

Table 6-53 A-CNCC Core Access Token Request

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

Table 6-53 (Cont.) A-CNCC Core Access Token Request

Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token"} Group by user: (CNE 1.8 or older) 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"}) Group by user: (OCCNE 1.9.x and higher versions) 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"})
Dimensions	- Host - HttpVersion - InstanceIdentifier - Method - Route_path - Scheme - Status - ResourcePath - ResourceType - UserId - UserName - AuthenticationType
Metric Type	Counter

6.3.8 A-CNCC Core Access Token Granted Responses

Table 6-54 A-CNCC Core Access Token Granted Responses

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

Table 6-54 (Cont.) A-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 or older)</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="200 OK"})</code> <i>Group by user: (OCCNE 1.9.x and higher 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="200 OK"})</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

6.3.9 A-CNCC Core Access Token Not Granted Responses

(Required) <Enter a short description here.>

Table 6-55 A-CNCC Core Access Token Not Granted Responses

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

Table 6-55 (Cont.) A-CNCC Core Access Token Not Granted Responses

Metric Filter	oc_ingressgateway_http_responses_total{InstanceIdIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status=~"4.* 5.*"} <i>Group by user: (CNE 1.8 or older)</i> sum by(Status,Method,kubernetes_namespace,ResourceType,UserId,UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status=~"4.* 5.*"}) <i>Group by user: (OCCNE 1.9.x and higher versions)</i> sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdIdentifier=~".*iam_ingressgateway",ResourcePath="/cncc/auth/realms/cncc/protocol/openid-connect/token",Status=~"4.* 5.*"})
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

6.3.10 A-CNCC Core User Authorization Failure Responses

Table 6-56 A-CNCC Core User Authorization Failure Responses

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

Table 6-56 (Cont.) A-CNCC Core User Authorization Failure Responses

Metric Filter	<code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status="403 FORBIDDEN", ResourceType!="UNKNOWN"}</code> <i>Group by user: (CNE 1.8 or older)</i> <code>sum by(Status,Method,kubernetes_namespace,ResourceType,UserId,UserName,kubernetes_pod_name,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status="403 FORBIDDEN", ResourceType!="UNKNOWN"})</code> <i>Group by user: (OCCNE 1.9.x and higher versions)</i> <code>sum by(Status,Method,namespace,ResourceType,UserId,UserName,pod,Scheme) (oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status="403 FORBIDDEN", ResourceType!="UNKNOWN"})</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

6.3.11 A-CNCC Core SCP Requests

Table 6-57 A-CNCC Core SCP Requests

Metric Details	Total number of requests received by A-CNCC Core for SCP
Metric Filter	<code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/ocscp/.+"}</code>
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

6.3.12 A-CNCC Core SCP Responses

Table 6-58 A-CNCC Core SCP Responses

Metric Details	Total number of responses sent by A-CNCC Core for SCP
Metric Filter	Single Route_path: oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_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

6.3.13 A-CNCC Core NRF Requests

Table 6-59 A-CNCC Core NRF Requests

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

6.3.14 A-CNCC Core NRF Responses

Table 6-60 A-CNCC Core NRF Responses

Metric Details	Total number of responses sent by A-CNCC Core for NRF
Metric Filter	Single Route_path: <code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/nrf-configuration/v1/*"}</code> Multiple Route_path: <code>oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_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

6.3.15 A-CNCC Core UDR Requests

Table 6-61 A-CNCC Core UDR Requests

Metric Details	Total number of requests received by A-CNCC Core for UDR
Metric Filter	Single Route_path: <code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/nudr-config/*"}</code> Multiple Route_path: <code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_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/*"}</code>

Table 6-61 (Cont.) A-CNCC Core UDR Requests

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

6.3.16 A-CNCC Core UDR Responses

Table 6-62 A-CNCC Core UDR Responses

Metric Details	Total number of responses sent by A-CNCC Core for UDR
Metric Filter	Single Route_path: oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/nudr-dr-prov/.*"} Multiple Route_path: oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_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/.*"}
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

6.3.17 A-CNCC Core POLICY Requests

Table 6-63 A-CNCC Core POLICY Requests

Metric Details	Total number of requests received by A-CNCC Core for POLICY
----------------	---

Table 6-63 (Cont.) A-CNCC Core POLICY Requests

Metric Filter	Single Route_path: <code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",ResourcePath=~".*/policyapi/*."}}</code> Multiple Route_path: <code>oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",ResourcePath=~".*/policyapi/*.*/oc-cnpolicy-configuration/*.*/pcf/*."}}</code>
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • UserName • AuthenticationType
Metric Type	Counter

6.3.18 A-CNCC Core POLICY Responses

Table 6-64 A-CNCC Core POLICY Responses

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

6.3.19 A-CNCC Core BSF Requests

Table 6-65 A-CNCC Core BSF Requests

Metric Details	Total number of requests received by A-CNCC Core for BSF
Metric Filter	<code>oc_ingressgateway_http_requests_total{InstanceIdIdentifier=~".*acore_ingressgateway",ResourcePath=~".*/bsfapi/.*"}</code> Multiple Route_path: <code>oc_ingressgateway_http_requests_total{InstanceIdIdentifier=~".*acore_ingressgateway",ResourcePath=~".*/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*"}</code>
Dimensions	• Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • Method • ResourcePath • ResourceType • UserId • Username • AuthenticationType
Metric Type	Counter

6.3.20 A-CNCC Core BSF Responses

Table 6-66 A-CNCC Core BSF Responses

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

6.3.21 A-CNCC Core SEPP Requests

Table 6-67 A-CNCC Core SEPP Requests

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

6.3.22 A-CNCC Core SEPP Responses

Table 6-68 A-CNCC Core SEPP Responses

Metric Details	Total number of responses sent by A-CNCC Core for SEPP
Metric Filter	<code>oc_ingressgateway_http_responses_total{InstanceIdIdentifier=~".*acore_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/sepp-configuration/.*"}</code> Multiple Route_path: <code>oc_ingressgateway_http_responses_total{InstanceIdIdentifier=~".*acore_ingressgateway",Status="200 OK",Method="GET",Route_path=~".*/sepp-configuration/.*/sepp/.*"}</code>
Dimensions	• Host • HttpVersion • InstanceIdentifier • Method • Route_path • Scheme • Status • ResourcePath • ResourceType • UserId • Username • AuthenticationType
Metric Type	Counter

6.3.23 A-CNCC Core NSSF Requests

Table 6-69 A-CNCC Core NSSF Requests

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

6.3.24 A-CNCC Core NSSF Responses

Table 6-70 A-CNCC Core NSSF Responses

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

6.3.25 A-CNCC Core DD Requests

Table 6-71 A-CNCC Core DD Requests

Metric Details	Total number of requests received by A-CNCC Core for DD
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/ocnadd/.*/ocnaddapi/.*"}

Table 6-71 (Cont.) A-CNCC Core DD Requests

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

6.3.26 A-CNCC Core DD Responses

Table 6-72 A-CNCC Core DD Responses

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

6.3.27 A-CNCC Core PROVGW Requests

Table 6-73 A-CNCC Core PROVGW Requests

Metric Details	Total number of requests received by A-CNCC Core for PROVGW
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/.*provgw-config.*/*provgw.*"}

Table 6-73 (Cont.) A-CNCC Core PROVGW Requests

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

6.3.28 A-CNCC Core PROVGW Responses

Table 6-74 A-CNCC Core PROVGW Responses

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

6.3.29 A-CNCC Core NWDAF Requests

Table 6-75 A-CNCC Core NWDAF Requests

Metric Details	Total number of requests received by A-CNCC Core for NWDAF
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*ocnwdaf.*.*ocnwdafapi.*"}

Table 6-75 (Cont.) A-CNCC Core NWDAF Requests

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

6.3.30 A-CNCC Core NWDAF Responses

Table 6-76 A-CNCC Core NWDAF Responses

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

6.3.31 A-CNCC Core cnDBTier Requests

Table 6-77 A-CNCC Core cnDBTier Requests

Metric Details	Total number of requests received by A-CNCC Core for cnDBTier
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*ocdbtier.*"}

Table 6-77 (Cont.) A-CNCC Core cnDBTier Requests

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

6.3.32 A-CNCC Core cnDBTier Responses

Table 6-78 A-CNCC Core cnDBTier Responses

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

6.3.33 A-CNCC Core OCCM Requests

Table 6-79 A-CNCC Core OCCM Requests

Metric Details	Total number of requests received by A-CNCC Core for OCCM
Metric Filter	oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*occm-config.*"}

Table 6-79 (Cont.) A-CNCC Core OCCM Requests

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

6.3.34 A-CNCC Core OCCM Responses

Table 6-80 A-CNCC Core OCCM Responses

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

7

CNC Console Alerts

This section provides information about CNC Console Alerts.

Note

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

- Review the `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.

7.1 CNC Console IAM Alerts

This section provides information about CNC Console IAM Alerts.

7.1.1 CncclamTotalIngressTrafficRateAboveMinorThreshold

Table 7-1 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 <code>occncc_alertrules_<version>.yaml</code> is when CNCC IAM Ingress Rate crosses 70 % of 1000 (Maximum ingress request rate)
Severity	Minor

Table 7-1 (Cont.) CnccIamTotalIngressTrafficRateAboveMinorThreshold

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>CnccIamTotalIngressTrafficRateAboveMajorThreshold</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.

7.1.2 CnccIamTotalIngressTrafficRateAboveMajorThreshold

Table 7-2 CnccIamTotalIngressTrafficRateAboveMajorThreshold

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
Alert details provided	Description: 'CNCC IAM Ingress traffic Rate is above the configured major threshold i.e. 800 requests per second (current value is: {{ \$value }})' For CNE 1.9.0 or later versions: summary: 'namespace: {{\$labels.namespace}}, podname: {{\$labels.pod}}, timestamp: {{ with query "time()" }}{ . first value humanizeTimestamp }}{{ end }}: Traffic Rate is above 80 Percent of Max requests per second(1000)' For CNE 1.8.x or previous versions: summary: 'namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{ . first value humanizeTimestamp }}{{ end }}: Traffic Rate is above 80 Percent of Max requests per second(1000)'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7001

Table 7-2 (Cont.) CncclamTotalIngressTrafficRateAboveMajorThreshold

Metric Used	oc_ingressgateway_http_requests_total
Resolution	The alert is cleared when the total Ingress Traffic rate falls below the Major threshold or when the total traffic rate crosses the Critical threshold, in which case the <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.

7.1.3 CncclamTotalIngressTrafficRateAboveCriticalThreshold

Table 7-3 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)'
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>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.

7.1.4 CncclamMemoryUsageCrossedMinorThreshold

Table 7-4 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, kube_pod_container_resource_limits Note:This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use a similar metric as exposed by the monitoring system.
Resolution	The alert gets cleared when the memory utilization falls below the Minor Threshold or crosses the major threshold, in which case <i>CncclamMemoryUsageCrossedMajorThreshold</i> 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.

7.1.5 CncclamMemoryUsageCrossedMajorThreshold

Table 7-5 CncclamMemoryUsageCrossedMajorThreshold

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

Table 7-5 (Cont.) CncclamMemoryUsageCrossedMajorThreshold

Resolution	The alert gets cleared when the memory utilization falls below the Major Threshold or crosses the critical threshold, in which case <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.
------------	--

7.1.6 CncclamMemoryUsageCrossedCriticalThreshold

Table 7-6 CncclamMemoryUsageCrossedCriticalThreshold

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

7.1.7 CncclamTransactionErrorRateAbove0.1Percent

Table 7-7 CncclamTransactionErrorRateAbove0.1Percent

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

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

Resolution	The alert is cleared when the number of failed transactions is below 0.1% of the total transactions or when the number of failed transactions crosses the 1% threshold in which case the <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.
------------	---

7.1.8 CncclamTransactionErrorRateAbove1Percent

Table 7-8 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.

7.1.9 CncclamTransactionErrorRateAbove10Percent

Table 7-9 CncclamTransactionErrorRateAbove10Percent

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

Table 7-9 (Cont.) CncclamTransactionErrorRateAbove10Percent

Resolution	The alert is cleared when the number of failed transactions is below 10% of the total transactions or when the number of failed transactions crosses the 25% threshold in which case the <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.
------------	--

7.1.10 CncclamTransactionErrorRateAbove25Percent

Table 7-10 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.

7.1.11 CncclamTransactionErrorRateAbove50Percent

Table 7-11 CncclamTransactionErrorRateAbove50Percent

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

Table 7-11 (Cont.) CnccIamTransactionErrorRateAbove50Percent

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

7.1.12 CnccIamIngressGatewayServiceDown

Table 7-12 CnccIamIngressGatewayServiceDown

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

7.1.13 CncclamFailedLogin

Table 7-13 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.

7.1.14 AdminUserCreation

Table 7-14 AdminUserCreation

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

Table 7-14 (Cont.) AdminUserCreation

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

7.1.15 CncclamAccessTokenFailure

Table 7-15 CncclamAccessTokenFailure

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

7.2 CNC Console Core Alerts

This section provides the information about CNC Console Core Alerts.

7.2.1 CnccCoreTotalIngressTrafficRateAboveMinorThreshold

Table 7-16 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 <code>cncc_alert_rules.yaml</code> 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.kubernetes_pod_name}}, 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 <code>occncc_alertrules_<version>.yaml</code> file. Steps: 1. Reassess why the CNCC Core is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

7.2.2 CnccCoreTotalIngressTrafficRateAboveMajorThreshold

Table 7-17 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 <code>cncc_alert_rules.yaml</code> is when CNCC Core Ingress Rate crosses 80 % of 1000 (Maximum ingress request rate)
Severity	Major

Table 7-17 (Cont.) CnccCoreTotalIngressTrafficRateAboveMajorThreshold

Alert details provided	Description: 'CNCC Core Ingress traffic Rate is above the configured major threshold i.e. 800 requests per second (current value is: {{ \$value }})' For CNE 1.9.0 or later versions: summary: 'namespace: {{\$labels.namespace}}, podname: {{\$labels.pod}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic Rate is above 80 Percent of Max requests per second(1000)' For CNE 1.8.x or previous versions: summary: 'namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic Rate is above 80 Percent of Max requests per second(1000)'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8001
Metric Used	oc_ingressgateway_http_requests_total
Resolution	The alert is cleared when the total Ingress Traffic rate falls below the Major threshold or when the total traffic rate crosses the Critical threshold, in which case the CnccCoreTotalIngressTrafficRate Above CriticalThreshold. 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.

7.2.3 CnccCoreTotalIngressTrafficRateAboveCriticalThreshold

Table 7-18 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 <i>cncc_alert_rules.yaml</i> is when CNCC Core Ingress Rate crosses 90 % of 1000 (Maximum ingress request rate)
Severity	Critical
Alert details provided	Description: 'CNCC Core Ingress traffic Rate is above the configured critical threshold i.e. 900 requests per second (current value is: {{ \$value }})' For CNE 1.9.0 or later versions: summary: 'namespace: {{\$labels.namespace}}, podname: {{\$labels.pod}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic Rate is above 90 Percent of Max requests per second(1000)' For CNE 1.8.x or previous versions: summary: 'namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic Rate is above 90 Percent of Max requests per second(1000)'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8001
Metric Used	oc_ingressgateway_http_requests_total

Table 7-18 (Cont.) CnccCoreTotalIngressTrafficRateAboveCriticalThreshold

Resolution	The alert is cleared when the Ingress Traffic rate falls below the Critical threshold. Note: The threshold is configurable in the <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.
------------	--

7.2.4 CnccCoreMemoryUsageCrossedMinorThreshold

Table 7-19 CnccCoreMemoryUsageCrossedMinorThreshold

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

7.2.5 CnccCoreMemoryUsageCrossedMajorThreshold

Table 7-20 CnccCoreMemoryUsageCrossedMajorThreshold

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

Table 7-20 (Cont.) CnccCoreMemoryUsageCrossedMajorThreshold

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

7.2.6 CnccCoreMemoryUsageCrossedCriticalThreshold

Table 7-21 CnccCoreMemoryUsageCrossedCriticalThreshold

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

7.2.7 CnccCoreTransactionErrorRateAbove0.1Percent

Table 7-22 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.

7.2.8 CnccCoreTransactionErrorRateAbove1Percent

Table 7-23 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.

7.2.9 CnccCoreTransactionErrorRateAbove10Percent

Table 7-24 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.

7.2.10 CnccCoreTransactionErrorRateAbove25Percent

Table 7-25 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.

7.2.11 CnccCoreTransactionErrorRateAbove50Percent

Table 7-26 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.

7.2.12 CnccCoreIngressGatewayServiceDown

Table 7-27 CnccCoreIngressGatewayServiceDown

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

Table 7-27 (Cont.) CnccCoreIngressGatewayServiceDown

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

7.2.13 CnccCoreFailedLogin

Table 7-28 CnccCoreFailedLogin

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

7.2.14 CnccCoreUnauthorizedAccess

Table 7-29 CnccCoreUnauthorizedAccess

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

Table 7-29 (Cont.) CnccCoreUnauthorizedAccess

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

7.2.15 CnccCoreAccessTokenFailure

Table 7-30 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 is required, Contact My Oracle Support.

7.3 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 `occncc_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
```

7.4 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

7.5 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.

7.6 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
```

7.7 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.

8

CNC Console KPIs

This section provides the information about CNC Console KPIs:

8.1 CNC Console IAM KPIs

This section provides the information about CNC Console IAM KPIs:

8.1.1 M-CNCC IAM Requests

Table 8-1 M-CNCC IAM Requests

Description	CNCC IAM Requests
Expression	For CNE 1.8 or older: all: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",kubernetes_namespace="\$namespace"}[2m])) For OCCNE 1.9.x and higher versions: all:sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*iam_ingressgateway",namespace="\$namespace"}[2m]))

8.1.2 M-CNCC IAM Responses

Table 8-2 M-CNCC IAM Responses

Description	M-CNCC IAM Responses
-------------	----------------------

Table 8-2 (Cont.) M-CNCC IAM Responses

Expression	For CNE 1.8 or older: - 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 OCCNE 1.9.x and higher 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]))
------------	--

8.1.3 M-CNCC IAM Success Rate

Table 8-3 M-CNCC IAM Success Rate

Description	M-CNCC IAM Success Rate (2xx responses divided by total responses)
-------------	--

Table 8-3 (Cont.) M-CNCC IAM Success Rate

Expression	For CNE 1.8 or older: all: sum(increase(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*iam_ingressgateway",Status=~"2.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*iam_ingressgateway",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m]))*100 For OCCNE 1.9.x and higher versions: all: sum(increase(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*iam_ingressgateway",Status=~"2.*",Route_path=~".*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*iam_ingressgateway",Route_path=~".*",namespace="\$namespace"}[5m]))*100
------------	---

8.1.4 M-CNCC IAM Error Rate

Table 8-4 M-CNCC IAM Error Rate

Description	M-CNCC IAM Error Rate (4xx or 5xx responses divided by total responses) for all as well as specific NFs
Expression	For CNE 1.8 or older: all: sum(increase(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*iam_ingressgateway",Status=~"4.* 5.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*iam_ingressgateway",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m]))*100 For OCCNE 1.9.x and higher versions: all: sum(increase(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*iam_ingressgateway",Status=~"4.* 5.*",Route_path=~".*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{Route_path=~".*",namespace="\$namespace"}[5m]))*100

8.2 M-CNCC Core KPIs

This section provides the information about M-CNCC Core KPIs:

8.2.1 M-CNCC Core Requests

Table 8-5 M-CNCC Core Requests

Description	M-CNCC Core Requests for NFs and specific NFs
-------------	---

Table 8-5 (Cont.) M-CNCC Core Requests

Expression	For CNE 1.8 or older: all: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",kubernetes_namespace="\$namespace"}[2m]))</code> For OCCNE 1.9.x and higher versions: - all: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",namespace="\$namespace"}[2m]))</code> - scp: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ocscp/.*" } [2m]))</code> - nrf: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/nrf-configuration/v1.*.*nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*" } [2m]))</code> - udr: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*" } [2m]))</code> - policy: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/policyapi/.*/oc-cnpolicy-configuration/.*/pcf/.*" } [2m]))</code> - bsf: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/bsfapi.*.*oc-bsf-configuration/.*/bsf/.*" } [2m]))</code> - sepp: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/sepp-configuration/.*/sepp/.*" } [2m]))</code> - nssf: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/nssf-configuration/.*/nssf/.*" } [2m]))</code> - dd: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ocnadd/.*/ocnaddapi/.*" } [2m]))</code> - provgw: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*provgw-config.*.*provgw.*" } [2m]))</code> - nwdaf: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*ocnwdaf.*.*ocnwdafapi.*" } [2m]))</code> - cndbtier: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*ocdbtier.*" } [2m]))</code>
------------	--

Table 8-5 (Cont.) M-CNCC Core Requests

	• <code>occm:</code> <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*occm-config.*"} [2m]))</code>
--	--

8.2.2 M-CNCC Core Responses

Table 8-6 M-CNCC Core Responses

Description	M-CNCC Core Responses for all as well as specific NFs
-------------	---

Table 8-6 (Cont.) M-CNCC Core Responses

Expression	For CNE 1.8 or older
	- all_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_error: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_4xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"4.*",ResourcePath=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_5xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"5.*",ResourcePath=~".*",kubernetes_namespace="\$namespace"}[5m])) - scp_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/ocscp/*",kubernetes_namespace="\$namespace"}[5m])) - nrf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/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=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/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=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/policyapi/* */oc-cnpolicy-configuration/* */pcf/*",kubernetes_namespace="\$namespace"}[5m])) - bsf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/bsfapi/* */oc-bsf-configuration/* */bsf/*",kubernetes_namespace="\$namespace"}[5m])) - sepp_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/sepp-configuration/* */sepp/*",kubernetes_namespace="\$namespace"}[5m])) - nssf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*",kubernetes_namespace="\$namespace"}[5m]))

Table 8-6 (Cont.) M-CNCC Core Responses

	<pre> Path=~"/nsssf-configuration/.*/nsssf/.*/kubernetes_namespace="\$namespace"}[5m])) • dd_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~"/ocnadd/.*/ocnaddapi/.*/kubernetes_namespace="\$namespace"}[5m])) • provgw_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*provgw-config.*provgw.*",kubernetes_namespace="\$namespace"}[5m])) • nwdaf_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*ocnwdaf.*ocnwdafapi.*",kubernetes_namespace="\$namespace"}[5m])) • cndbtier_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*ocdbtier.*",kubernetes_namespace="\$namespace"}[5m])) • occm_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*occm-config.*",kubernetes_namespace="\$namespace"}[5m])) </pre> For OCCNE 1.9.x and higher versions: - all_success: <pre>sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*",namespace="\$namespace"}[5m]))</pre> - all_error:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"4.*5.*",ResourcePath=~".*",namespace="\$namespace"}[5m])) - all_4xx: <pre>sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"4.*",ResourcePath=~".*",namespace="\$namespace"}[5m]))</pre> - all_5xx: <pre>sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"5.*",ResourcePath=~".*",namespace="\$namespace"}[5m]))</pre> - scp_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~"/ocscp/.*",namespace="\$namespace"}[5m])) - nrf_success: <pre>sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~"/nrf-configuration/v1.*nrf-state-data.*ocnrf-swagger.*nrf-status-data.*nrf/nf-common-component.*",namespace="\$namespace"}[5m]))</pre> - udr_success: <pre>sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~"/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]))</pre>
--	---

Table 8-6 (Cont.) M-CNCC Core Responses

	• policy_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/policyapi/.*/.*oc-cnpolicy-configuration/.*/.*pcf/.*",namespace="\$namespace"}[5m])) • bsf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/bsfapi/.*/.*oc-bsf-configuration/.*/.*bsf/.*",namespace="\$namespace"}[5m])) • sepp_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/sepp-configuration/.*/.*sepp/.*",namespace="\$namespace"}[5m])) • nssf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/nssf-configuration/.*/.*nssf/.*",namespace="\$namespace"}[5m])) • dd_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/ocnadd/.*/.*ocnaddapi/.*",namespace="\$namespace"}[5m])) • provgw_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/provgw-config/.*/.*provgw.*",namespace="\$namespace"}[5m])) • nwdaf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/ocnwdaf/.*/.*ocnwdafapi.*",namespace="\$namespace"}[5m])) • cndbtier_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/ocdbtier.*",namespace="\$namespace"}[5m])) • occm_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/occm-config.*",namespace="\$namespace"}[5m]))
--	--

8.2.3 M-CNCC Core Success Rate

Table 8-7 M-CNCC Core Success Rate

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

Table 8-7 (Cont.) M-CNCC Core Success Rate

Expression	For CNE 1.8 or older:
	- all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"2.*", ResourcePath}=\sim\text{".*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", ResourcePath}=\sim\text{".*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - scp: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"2.*", ResourcePath}=\sim\text{".*/ocscp/.*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", ResourcePath}=\sim\text{"/ocscp/.*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - nrf: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"2.*", ResourcePath}=\sim\text{"/nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", ResourcePath}=\sim\text{"/nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - udr: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"2.*", ResourcePath}=\sim\text{"/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", ResourcePath}=\sim\text{"/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - policy: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"2.*", ResourcePath}=\sim\text{"/policyapi/.*/oc-cnppolicy-configuration/.*/pcf/.*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", ResourcePath}=\sim\text{"/policyapi/.*/oc-cnppolicy-configuration/.*/pcf/.*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - bsf: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"2.*", ResourcePath}=\sim\text{"/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"2.*", ResourcePath}=\sim\text{"/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*", kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$

Table 8-7 (Cont.) M-CNCC Core Success Rate

	<pre>sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",ResourcePath=~".*/ bsfapi.*"/oc-bsf- configuration/.*/.*bsf/.*,kubernetes_namespace="\$namespac e"}[5m]))*100</pre> <pre>sepp: sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*/sepp-configuration/.*/.* sepp/.*,kubernetes_namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",ResourcePath=~".*/sepp- configuration/.*/.* sepp/.*,kubernetes_namespace="\$namespace"}[5m]))*100</pre> <pre>nssf: sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*/nnsf-configuration/.*/.* nssf/.*,kubernetes_namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",ResourcePath=~".*/nnsf- configuration/.*/.* nssf/.*,kubernetes_namespace="\$namespace"}[5m]))*100</pre> <pre>dd: sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*/ocnadd/.*/.* ocnaddapi/.*,kubernetes_namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",ResourcePath=~".*/ ocnadd/.*/.* ocnaddapi/.*,kubernetes_namespace="\$namespace"} [5m]))*100</pre> <pre>provgw: sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*provgw- config.*/*provgw.*,kubernetes_namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",ResourcePath=~".*provg w- config.*/*provgw.*,kubernetes_namespace="\$namespace"} [5m]))*100</pre> <pre>nwdaf: sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*ocnwdaf.*/*ocnwdafapi.*,kubernetes_namespace="\$n amespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",ResourcePath=~".*ocnwd af.*/*ocnwdafapi.*,kubernetes_namespace="\$namespace"} [5m]))*100</pre> <pre>cndbtier: sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*ocdbtier.*,kubernetes_namespace="\$namespace"} [5m]))/</pre>
--	---

Table 8-7 (Cont.) M-CNCC Core Success Rate

	sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*ocdbti er.*",kubernetes_namespace="\$namespace"}[5m]))*100 - occm: sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*occm- config.*",kubernetes_namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*occm- config.*",kubernetes_namespace="\$namespace"}[5m]))*100 For OCCNE 1.9.x and higher versions: - all: sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*",namespace="\$namespace"}[5m]))*100 - scp: sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/ocscp/.*",namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ocscp/.*",namespace="\$namespace"}[5m]))*100 - nrf: sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/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{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*",namespace="\$namespace"}[5m]))*100 - udr: sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/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{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/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 - policy: sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePath=~".*/policyapi/.*/oc-cnpolicy-configuration/.*/pcf/.*",namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/policyapi/.*/oc-cnpolicy-
--	---

Table 8-7 (Cont.) M-CNCC Core Success Rate

	<pre>configuration/.*/pcf/.*/,namespace="\$namespace"} [5m]))*100</pre> - bsf: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*/bsfapi/.*/oc-bsf- configuration/.*/bsf/.*/,namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",ResourcePath=~".*/ bsfapi/.*/oc-bsf- configuration/.*/bsf/.*/,namespace="\$namespace"} [5m]))*100</pre> - sepp: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*/sepp-configuration/.*/.*/ sepp/.*/,namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",ResourcePath=~".*/sepp- configuration/.*/sepp/.*/,namespace="\$namespace"} [5m]))*100</pre> - nssf: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*/nssf-configuration/.*/.*/ nssf/.*/,namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",ResourcePath=~".*/nssf- configuration/.*/nssf/.*/,namespace="\$namespace"} [5m]))*100</pre> - dd: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*/ocnadd/.*/ocnaddapi/.*/,namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",ResourcePath=~".*/ ocnadd/.*/ocnaddapi/.*/,namespace="\$namespace"} [5m]))*100</pre> - provgw: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*/provgw- config/.*/provgw.*/,namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",ResourcePath=~".*/provg w-config/.*/provgw.*/,namespace="\$namespace"}[5m]))*100</pre> - nwdaf: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*/ocnwdaf.*/ocnwdafapi.*/,namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",ResourcePath=~".*/ocnwd af.*/ocnwdafapi.*/,namespace="\$namespace"}[5m]))*100</pre> - cndbtier: <pre>sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa</pre>
--	---

Table 8-7 (Cont.) M-CNCC Core Success Rate

	<pre>th=~".*ocdbtier.*",namespace="\$namespace"){5m}))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceI dentifier=~".*mcore_ingressgateway",ResourcePath=~".*ocdbti er.*",namespace="\$namespace">{5m}))*100</pre> • <pre>occm: sum(rate(oc_ingressgateway_http_responses_total{InstanceI dentifier=~".*mcore_ingressgateway",Status=~"2.*",ResourcePa th=~".*occm-config.*",namespace="\$namespace">{5m}))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceI dentifier=~".*mcore_ingressgateway",ResourcePath=~".*occm- config.*",namespace="\$namespace">{5m}))*100</pre>
--	---

8.2.4 M-CNCC Core Error Rate

Table 8-8 M-CNCC Core Error Rate

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

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

Expression	For CNE 1.8 or older:
	- all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"4.* 5.*", ResourcePath}=\sim\text{".*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", ResourcePath}=\sim\text{".*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - scp: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"4.* 5.*", ResourcePath}=\sim\text{".*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", ResourcePath}=\sim\text{".*/ocscp/.*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - nrf: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"4.* 5.*", ResourcePath}=\sim\text{".*/nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", ResourcePath}=\sim\text{".*/nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - udr: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"4.* 5.*", ResourcePath}=\sim\text{".*/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", ResourcePath}=\sim\text{".*/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - policy: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"4.* 5.*", ResourcePath}=\sim\text{".*/policyapi/.*/oc-cnppolicy-configuration/.*/pcf/.*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", ResourcePath}=\sim\text{".*/policyapi/.*/oc-cnppolicy-configuration/.*/pcf/.*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - bsf: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"4.* 5.*", ResourcePath}=\sim\text{".*/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*mcore_ingressgateway", Status}=\sim\text{"4.* 5.*", ResourcePath}=\sim\text{".*/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*"}\}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$

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

	<pre>e"}{5m}))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ bsfapi.*.*/*oc-bsf- configuration/*.*/*bsf/*.*,kubernetes_namespace="\$namespac e"}{5m}))*100</pre> <pre>sepp: sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*/sepp-configuration/*.*/* sepp/*.*,kubernetes_namespace="\$namespace"}{5m}))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ sepp-configuration/*.*/* sepp/*.*,kubernetes_namespace="\$namespace"}{5m}))*100</pre> <pre>nssf: sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*/nssf-configuration/*.*/* nssf/*.*,kubernetes_namespace="\$namespace"}{5m}))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ nssf-configuration/*.*/* nssf/*.*,kubernetes_namespace="\$namespace"}{5m}))*100</pre> <pre>dd: sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*/ocnadd/*.*/* ocnaddapi/*.*,kubernetes_namespace="\$namespace"}{5m}))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ ocnadd/*.*/* ocnaddapi/*.*,kubernetes_namespace="\$namespace"} {5m}))*100</pre> <pre>provgw: sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*provgw- config.*.*provgw.*,kubernetes_namespace="\$namespace"} {5m}))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*p rovgw- config.*.*provgw.*,kubernetes_namespace="\$namespace"} {5m}))*100</pre> <pre>nwdaf: sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*ocnwdaf.*.*ocnwdafapi.*,kubernetes_ namespace="\$namespace"}{5m}))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*o cnwdaf.*.*ocnwdafapi.*,kubernetes_namespace="\$namespa ce"}{5m}))*100</pre> <pre>cndbtier: sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*ocdbtier.*.*,kubernetes_namespace="\$</pre>
--	--

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

	<pre>namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*o cldbtierr.*",kubernetes_namespace="\$namespace"}[5m]))*100</pre> - occm: <pre>sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~".*4.* 5.*",ResourcePath=~".*occm- config.*",kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*o ccm-config.*",kubernetes_namespace="\$namespace"} [5m]))*100</pre> For OCCNE 1.9.x and higher versions: - all: <pre>sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~".*4.* 5.*",ResourcePath=~".*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*", namespace="\$namespace"}[5m]))*100</pre> - scp: <pre>sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~".*4.* 5.*",ResourcePath=~".*/soothsayer/v1/*.* ocscp/*.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ ocscp/*.*",namespace="\$namespace"}[5m]))*100</pre> - nrf: <pre>sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~".*4.* 5.*",ResourcePath=~".*/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{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ nrf-configuration/v1/*.* nrf-state-data/*.* ocnrf-swagger/*.* nrf-status-data/*.* nrf/nf-common- component/*.*",namespace="\$namespace"}[5m]))*100</pre> - udr: <pre>sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~".*4.* 5.*",ResourcePath=~".*/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{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ 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> - policy: <pre>sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~".*4.* 5.*",ResourcePath=~".*/policyapi/*.* oc-cnpolicy- configuration/*.* pcf/*.*",namespace="\$namespace"}[5m]))/</pre>
--	---

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

	<pre>sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/policyapi/.*/.*oc-cnpolicy-configuration/.*/pcf/.*",namespace="\$namespace"}[5m]))*100</pre> <pre>bsf: sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*/bsfapi/.*/.*oc-bsf-configuration/.*/bsf/.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/bsfapi/.*/.*oc-bsf-configuration/.*/bsf/.*",namespace="\$namespace"}[5m]))*100</pre> <pre>sepp: sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*/sepp-configuration/.*/.*sepp/.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/sepp-configuration/.*/.*sepp/.*",namespace="\$namespace"}[5m]))*100</pre> <pre>nssf: sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*/nssf-configuration/.*/.*nssf/.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/nssf-configuration/.*/.*nssf/.*",namespace="\$namespace"}[5m]))*100</pre> <pre>dd: sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*/ocnadd/.*/.*ocnaddapi/.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ocnadd/.*/.*ocnaddapi/.*",namespace="\$namespace"}[5m]))*100</pre> <pre>provgw: sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*/provgw-config.*.*provgw.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Route_path=~".*/provgw-config.*.*provgw.*",namespace="\$namespace"}[5m]))*100</pre> <pre>nwdaf: sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*/ocnwdaf.*.*ocnwdafapi.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*/ocnwdaf.*.*ocnwdafapi.*",namespace="\$namespace"}[5m]))*100</pre>
--	---

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

	cnwdaf.* .ocnwdafapi.*",namespace="\$namespace"} [5m]))*100 • cndbtier: sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*ocdbtier.*",namespace="\$namespace"} [5m]))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*o cdbtier.*",namespace="\$namespace"}[5m]))*100 • occm: sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",Status=~"4.* 5.*",ResourcePath=~".*occm- config.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{Insta ncelIdentifier=~".*mcore_ingressgateway",ResourcePath=~".*o ccm-config.*",namespace="\$namespace"}[5m]))*100
--	---

8.3 A-CNCC Core KPIs

This section provides the information about A-CNCC Core KPIs:

8.3.1 A-CNCC Core Requests

Table 8-9 A-CNCC Core Requests

Description	A-CNCC Core Requests for all as well as specific NFs
-------------	--

Table 8-9 (Cont.) A-CNCC Core Requests

Expression	For CNE 1.8 or older: all: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",kubernetes_namespace="\$namespace"}[2m]))</code> For OCCNE 1.9.x and higher versions: - all: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",namespace="\$namespace"}[2m]))</code> - scp: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/ocscp/*"}[2m]))</code> - nrf: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/nrf-configuration/v1.**/nrf-state-data/**/ocnrf-swagger/**/nrf-status-data/**/nrf/nf-common-component/*"} [2m]))</code> - udr: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/nudr-dr-prov/**/nudr-dr-mgm/**/nudr-group-id-map-prov/**/sif-group-prov/**/nudr-config/**/udr/nf-common-component/**/n5g-eir-prov/*"} [2m]))</code> - policy: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",ResourcePath=~".*/policyapi/**/oc-cnpolicy-configuration/**/pcf/*"} [2m]))</code> - bsf: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",ResourcePath=~".*/bsfapi/**/oc-bsf-configuration/**/bsf/*"} [2m]))</code> - sepp: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/sepp-configuration/**/sepp/*"} [2m]))</code> - nssf: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/nssf-configuration/**/nssf/*"} [2m]))</code> - dd: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/ocnadd/**/ocnaddapi/*"} [2m]))</code> - provgw: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/provgw-config/**/provgw/*"} [2m]))</code> - nwdaf: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/ocnwdaf/**/ocnwdafapi/*"} [2m]))</code> - cndbtier: <code>sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*/ocdbtier/*"} [2m]))</code>
------------	---

Table 8-9 (Cont.) A-CNCC Core Requests

	occm: sum(irate(oc_ingressgateway_http_requests_total{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*occm-config.*"} [2m]))
--	--

8.3.2 A-CNCC Core Responses

Table 8-10 A-CNCC Core Responses

Description	A-CNCC Core Responses for all as well as specific NFs
-------------	---

Table 8-10 (Cont.) A-CNCC Core Responses

Expression	For CNE 1.8 or older
	- all_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_error: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"4.* 5.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_4xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"4.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - all_5xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"5.*",Route_path=~".*",kubernetes_namespace="\$namespace"}[5m])) - scp_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/ocscp/.*",kubernetes_namespace="\$namespace"}[5m])) - nrf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_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=~".*acore_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=~".*acore_ingressgateway",Status=~"2.*",ResourcePath=~".*/policyapi/* */oc-cnpolicy-configuration/* */pcf/*",kubernetes_namespace="\$namespace"}[5m])) - bsf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",ResourcePath=~".*/bsfapi/* */oc-bsf-configuration/* */bsf/*",kubernetes_namespace="\$namespace"}[5m])) - sepp_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/sepp-configuration/* */sepp/*",kubernetes_namespace="\$namespace"}[5m])) - nssf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/nssf-configuration/* */nssf/*",kubernetes_namespace="\$namespace"}[5m]))

Table 8-10 (Cont.) A-CNCC Core Responses

	- dd_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/ocnadd/*.*/*ocnaddapi/*.*",kubernetes_namespace="\$namespace"}[5m])) - provgw_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*provgw-config.*.*",kubernetes_namespace="\$namespace"}[5m])) - nwdaf_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*ocnwdaf.*.*",kubernetes_namespace="\$namespace"}[5m])) - cndbtier_success:sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*ocdbtier.*.*",kubernetes_namespace="\$namespace"}[5m])) - occm_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*occm-config.*.*",kubernetes_namespace="\$namespace"}[5m])) For OCCNE 1.9.x and higher versions: - all_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*.*",namespace="\$namespace"}[5m])) - all_error: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"4.* 5.*",Route_path=~".*.*",namespace="\$namespace"}[5m])) - all_4xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"4.*",Route_path=~".*.*",namespace="\$namespace"}[5m])) - all_5xx: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"5.*",Route_path=~".*.*",namespace="\$namespace"}[5m])) - scp_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/ocscpl/*.*",namespace="\$namespace"}[5m])) - nrf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_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=~".*acore_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=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/policy/*.*",namespace="\$namespace"}[5m]))
--	---

Table 8-10 (Cont.) A-CNCC Core Responses

	dentifier=~".*acore_ingressgateway",Status=~"2.*",ResourcePath=~".*/policyapi/.*/oc-cnpolicy-configuration/.*/pcf/.*",namespace="\$namespace"){5m})) • bsf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",ResourcePath=~".*/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*",namespace="\$namespace">{5m})) • sepp_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/sepp-configuration/.*/sepp/.*",namespace="\$namespace">{5m})) • nssf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/nssf-configuration/.*/nssf/.*",namespace="\$namespace">{5m})) • dd_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/ocnadd/.*/ocnaddapi/.*",namespace="\$namespace">{5m})) • provgw_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/provgw-config/.*/provgw.*",namespace="\$namespace">{5m})) • nwdaf_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/ocnwdaf/.*/ocnwdafapi.*",namespace="\$namespace">{5m})) • cndbtier_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/ocdbtier.*",namespace="\$namespace">{5m})) • occm_success: sum(irate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*/occm-config.*",namespace="\$namespace">{5m}))
--	---

8.3.3 A-CNCC Core Success Rate

Table 8-11 A-CNCC Core Success Rate

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

Table 8-11 (Cont.) A-CNCC Core Success Rate

Expression	For CNE 1.8 or older:
	- all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"2.*", Route_path}=\sim\text{".*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{".*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - scp: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"2.*", Route_path}=\sim\text{".*"/ocscp/.*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{"/ocscp/.*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - nrf: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"2.*", Route_path}=\sim\text{"/nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{"/nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - udr: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"2.*", Route_path}=\sim\text{"/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{"/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - policy: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"2.*", ResourcePath}=\sim\text{"/policyapi/.*/oc-cnpolicy-configuration/.*/pcf/.*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", ResourcePath}=\sim\text{"/policyapi/.*/oc-cnpolicy-configuration/.*/pcf/.*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - bsf: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"2.*", ResourcePath}=\sim\text{"/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"2.*", ResourcePath}=\sim\text{"/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*"}, \text{kubernetes_namespace}=\text{"\$namespace"}\}[5m]))} * 100$

Table 8-11 (Cont.) A-CNCC Core Success Rate

	<pre>entifier=~".*acore_ingressgateway",ResourcePath=~".*/ bsfapi.*./oc-bsf- configuration/.*/.*/bsf/.**",kubernetes_namespace="\$namespac e")][5m]))*100</pre> <pre>sepp: sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Status=~"2.**",Route_path =~".*/sepp-configuration/.*/.*/ sepp/.**",kubernetes_namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Route_path=~".*/sepp- configuration/.*/.*/ sepp/.**",kubernetes_namespace="\$namespace"}[5m]))*100</pre> <pre>nssf: sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Status=~"2.**",Route_path =~".*/nssf-configuration/.*/.*/ nssf/.**",kubernetes_namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Route_path=~".*/nssf- configuration/.*/.*/ nssf/.**",kubernetes_namespace="\$namespace"}[5m]))*100</pre> <pre>dd: sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Status=~"2.**",Route_path =~".*/ocnadd/.*/.*/ ocnaddapi/.**",kubernetes_namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Route_path=~".*/ ocnadd/.*/.*/ ocnaddapi/.**",kubernetes_namespace="\$namespace"} [5m]))*100</pre> <pre>provgw: sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Status=~"2.**",Route_path =~".*/provgw- config.*./provgw.**",kubernetes_namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Route_path=~".*/provgw- config.*./provgw.**",kubernetes_namespace="\$namespace"} [5m]))*100</pre> <pre>nwdaf: sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Status=~"2.**",Route_path =~".*/ocnwdaf.*./ocnwdafapi.**",kubernetes_namespace="\$na mespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Route_path=~".*/ocnwdaf.* ./ocnwdafapi.**",kubernetes_namespace="\$namespace"} [5m]))*100</pre> <pre>cndbtier: sum(rate(oc_ingressgateway_http_responses_total{InstanceId entifier=~".*acore_ingressgateway",Status=~"2.**",Route_path =~".*/ocdbtier.**",kubernetes_namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceId</pre>
--	---

Table 8-11 (Cont.) A-CNCC Core Success Rate

	entifier=~".*acore_ingressgateway",Route_path=~".*ocdbtier.*",kubernetes_namespace="\$namespace"){5m}))*100 - occm: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*acore_ingressgateway",Status=~"2.*",Route_path=~".*occm-config.*",kubernetes_namespace="\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*acore_ingressgateway",Route_path=~".*occm-config.*",kubernetes_namespace="\$namespace"}\}[5m]))}*100$ For OCCNE 1.9.x and higher versions: - all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*",namespace="\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{Route_path=~".*",namespace="\$namespace"}\}[5m]))}*100$ - scp: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/ocscp.*",namespace="\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/ocscp.*",namespace="\$namespace"}\}[5m]))}*100$ - nrf: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/nrf-configuration/v1.*.**/nrf-state-data/*.**/ocnrf-swagger/*.**/nrf-status-data/*.**/nrf/nf-common-component/*.*",namespace="\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nrf-configuration/v1.*.**/nrf-state-data/*.**/ocnrf-swagger/*.**/nrf-status-data/*.**/nrf/nf-common-component/*.*",namespace="\$namespace"}\}[5m]))}*100$ - udr: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*/nudr-dr-prov/*.**/nudr-dr-mgm/*.**/nudr-group-id-map-prov/*.**/slf-group-prov/*.**/nudr-config/*.**/udr/nf-common-component/*.**/n5g-eir-prov/*.*",namespace="\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nudr-dr-prov/*.**/nudr-dr-mgm/*.**/nudr-group-id-map-prov/*.**/slf-group-prov/*.**/nudr-config/*.**/udr/nf-common-component/*.**/n5g-eir-prov/*.*",namespace="\$namespace"}\}[5m]))}*100$ - policy: $\frac{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",ResourcePath=~".*/policyapi/*.**/oc-cnpolicy-configuration/*.**/pcf/*.*",namespace="\$namespace"}\}[5m]))}{\text{sum}(\text{rate}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier=~".*core_ingressgateway",ResourcePath=~".*/policyapi/*.**/oc-cnpolicy-$
--	---

Table 8-11 (Cont.) A-CNCC Core Success Rate

	<pre>configuration/.*/pcf/.*/namespace="\$namespace"} [5m]))*100</pre> - bsf: <pre>sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Status=~"2.*",ResourcePath =~".*/bsfapi/.*/oc-bsf- configuration/.*/bsf/.*/namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",ResourcePath=~".*/ bsfapi/.*/oc-bsf- configuration/.*/bsf/.*/namespace="\$namespace"} [5m]))*100</pre> - sepp: <pre>sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Status=~"2.*",Route_path= =~".*/sepp-configuration/.*/ sepp/.*/namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Route_path=~".*/sepp- configuration/.*/sepp/.*/namespace="\$namespace"} [5m]))*100</pre> - nssf: <pre>sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Status=~"2.*",Route_path= =~".*/nssf-configuration/.*/ nssf/.*/namespace="\$namespace"}[5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Route_path=~".*/nssf- configuration/.*/nssf/.*/namespace="\$namespace"} [5m]))*100</pre> - dd: <pre>sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Status=~"2.*",Route_path= =~".*/ocnadd/.*/ocnaddapi/.*/namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Route_path=~".*/ ocnadd/.*/ocnaddapi/.*/namespace="\$namespace"} [5m]))*100</pre> - provgw: <pre>sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Status=~"2.*",Route_path= =~".*/provgw-config.*.*provgw.*",namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Route_path=~".*/provgw- config.*.*provgw.*",namespace="\$namespace"}[5m]))*100</pre> - nwdaf: <pre>sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Status=~"2.*",Route_path= =~".*/ocnwdaf.*.*ocnwdafapi.*",namespace="\$namespace"} [5m]))/ sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Route_path=~".*/ocnwdaf.*. *ocnwdafapi.*",namespace="\$namespace"}[5m]))*100</pre> - cndbtier: <pre>sum(rate(oc_ingressgateway_http_responses_total{Instanceid entifier=~".*core_ingressgateway",Status=~"2.*",Route_path=</pre>
--	---

Table 8-11 (Cont.) A-CNCC Core Success Rate

	~".*ocdbtier.*",namespace="\$namespace"){5m}))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*ocdbtier.*",namespace="\$namespace">{5m}})*100 • occm: sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"2.*",Route_path=~".*occm-config.*",namespace="\$namespace">{5m}}))/ sum(rate(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*occm-config.*",namespace="\$namespace">{5m}})*100
--	--

8.3.4 A-CNCC Core Error Rate

Table 8-12 A-CNCC Core Error Rate

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

Table 8-12 (Cont.) A-CNCC Core Error Rate

Expression	For CNE 1.8 or older:
	- all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", Route_path}=\sim\text{".*"}\text{,kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]}))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{".*"}\text{,kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]}))} \times 100$ - scp: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", Route_path}=\sim\text{".*"}\text{,kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]}))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{".*"}\text{,kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]}))} \times 100$ - nrf: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", Route_path}=\sim\text{".*"}\text{,nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*,kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]}))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{".*"}\text{,nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*,kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]}))} \times 100$ - udr: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", Route_path}=\sim\text{".*"}\text{,nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*,kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]}))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{".*"}\text{,nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*,kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]}))} \times 100$ - policy: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", ResourcePath}=\sim\text{".*"}\text{,policyapi/.*/oc-cnppolicy-configuration/.*/pcf/.*,kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]}))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", InstanceIdentifier}=\sim\text{".*acore_ingressgateway", ResourcePath}=\sim\text{".*"}\text{,policyapi/.*/oc-cnppolicy-configuration/.*/pcf/.*,kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]}))} \times 100$ - bsf: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", ResourcePath}=\sim\text{".*"}\text{,bsfapi/.*/oc-bsf-$

Table 8-12 (Cont.) A-CNCC Core Error Rate

	<pre>configuration/.*/.*/bsf/.*/kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",ResourcePath=~".*/bsfapi/.*/oc-bsf-configuration/.*/.*/bsf/.*/kubernetes_namespace="\$namespace"}[5m]))*100</pre> • sepp: <pre>sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"4.* 5.*",Route_path=~".*/sepp-configuration/.*/.*/sepp/.*/kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_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=~".*acore_ingressgateway",Status=~"4.* 5.*",Route_path=~".*/nssf-configuration/.*/.*/nssf/.*/kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/nssf-configuration/.*/.*/nssf/.*/kubernetes_namespace="\$namespace"}[5m]))*100</pre> • dd: <pre>sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"4.* 5.*",Route_path=~".*/ocnadd/.*/.*/ocnaddapi/.*/kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/ocnadd/.*/.*/ocnaddapi/.*/kubernetes_namespace="\$namespace"}[5m]))*100</pre> • provgw: <pre>sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"4.* 5.*",Route_path=~".*/provgw-config/.*/provgw.*",kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/provgw-config/.*/provgw.*",kubernetes_namespace="\$namespace"}[5m]))*100</pre> • nwdaf: <pre>sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",Status=~"4.* 5.*",Route_path=~".*/ocnwdaf.* ocnwdafapi.*",kubernetes_namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*acore_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*/ocnwdaf.* ocnwdafapi.*",kubernetes_namespace="\$namespace"}[5m]))*100</pre>
--	---

Table 8-12 (Cont.) A-CNCC Core Error Rate

	- cndbtier: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", Route_path}=\sim\text{".*ocdbtier.*", kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]})}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", InstanceIdentifier}=\sim\text{".*core_ingressgateway", Route_path}=\sim\text{".*ocdbtier.*", kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]})}) * 100$ - occm: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", Route_path}=\sim\text{".*occm-config.*", kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]})}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", InstanceIdentifier}=\sim\text{".*core_ingressgateway", Route_path}=\sim\text{".*occm-config.*", kubernetes_namespace}=\text{"\$namespace"}\}\text{[5m]})}) * 100$ For OCCNE 1.9.x and higher versions: - all: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", Route_path}=\sim\text{".*", namespace}=\text{"\$namespace"}\}\text{[5m]})}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{".*", namespace}=\text{"\$namespace"}\}\text{[5m]})}) * 100$ - scp: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", Route_path}=\sim\text{".*/soothsayer/v1/.*/ocscp/.*", namespace}=\text{"\$namespace"}\}\text{[5m]})}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{".*/ocscp/.*", namespace}=\text{"\$namespace"}\}\text{[5m]})}) * 100$ - nrf: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", Route_path}=\sim\text{".*/nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*", namespace}=\text{"\$namespace"}\}\text{[5m]})}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{".*/nrf-configuration/v1/.*/nrf-state-data/.*/ocnrf-swagger/.*/nrf-status-data/.*/nrf/nf-common-component/.*", namespace}=\text{"\$namespace"}\}\text{[5m]})}) * 100$ - udr: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.*"}\text{5.*", Route_path}=\sim\text{".*/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*", namespace}=\text{"\$namespace"}\}\text{[5m]})}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Route_path}=\sim\text{".*/nudr-dr-prov/.*/nudr-dr-mgm/.*/nudr-group-id-map-prov/.*/slf-group-prov/.*/nudr-config/.*/udr/nf-common-component/.*/n5g-eir-prov/.*", namespace}=\text{"\$namespace"}\}\text{[5m]})}) * 100$
--	---

Table 8-12 (Cont.) A-CNCC Core Error Rate

	- policy: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.* 5.*", ResourcePath}=\sim\text{"/policyapi/.*/oc-cnpolicy-configuration/.*/pcf/.*", namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", InstanceIdentifier}=\sim\text{".*core_ingressgateway", ResourcePath}=\sim\text{"/policyapi/.*/oc-cnpolicy-configuration/.*/pcf/.*", namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - bsf: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.* 5.*", ResourcePath}=\sim\text{"/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*", namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", InstanceIdentifier}=\sim\text{".*core_ingressgateway", ResourcePath}=\sim\text{"/bsfapi/.*/oc-bsf-configuration/.*/bsf/.*", namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - sepp: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.* 5.*", Route_path}=\sim\text{"/sepp-configuration/.*/sepp/.*", namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", InstanceIdentifier}=\sim\text{".*core_ingressgateway", Route_path}=\sim\text{"/sepp-configuration/.*/sepp/.*", namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - nssf: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.* 5.*", Route_path}=\sim\text{"/nssf-configuration/.*/nssf/.*", namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", InstanceIdentifier}=\sim\text{".*core_ingressgateway", Route_path}=\sim\text{"/nssf-configuration/.*/nssf/.*", namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - dd: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.* 5.*", Route_path}=\sim\text{"/ocnadd/.*/ocnaddapi/.*", namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", InstanceIdentifier}=\sim\text{".*core_ingressgateway", Route_path}=\sim\text{"/ocnadd/.*/ocnaddapi/.*", namespace}=\text{"\$namespace"}\}[5m]))} * 100$ - provgw: $\frac{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", Status}=\sim\text{"4.* 5.*", Route_path}=\sim\text{"/provgw-config/.*/provgw.*", namespace}=\text{"\$namespace"}\}[5m]))}{\text{sum}(\text{increase}(\text{oc_ingressgateway_http_responses_total}\{\text{InstanceIdentifier}=\sim\text{".*acore_ingressgateway", InstanceIdentifier}=\sim\text{".*core_ingressgateway", Route_path}=\sim\text{"/provgw-config/.*/provgw.*", namespace}=\text{"\$namespace"}\}[5m]))} * 100$
--	---

Table 8-12 (Cont.) A-CNCC Core Error Rate

	- nwdaf: sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*",Route_path=~".*ocnwdaf.* .ocnwdafapi.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*ocnwdaf.* .ocnwdafapi.*",namespace="\$namespace"}[5m]))*100 - cndbtier: sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*",Route_path=~".*ocdbtier.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*ocdbtier.*",namespace="\$namespace"}[5m]))*100 - occm: sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",Status=~"4.* 5.*",Route_path=~".*occm-config.*",namespace="\$namespace"}[5m]))/ sum(increase(oc_ingressgateway_http_responses_total{InstanceIdentifier=~".*core_ingressgateway",InstanceIdentifier=~".*core_ingressgateway",Route_path=~".*occm-config.*",namespace="\$namespace"}[5m]))*100
--	---

8.4 CNC Console Common KPIs

This section provides the information about CNC Console common KPIs:

8.4.1 CPU Usage

Table 8-13 CPU Usage

Description	CPU usage by all as well as individual microservices in CNCC
-------------	--

Table 8-13 (Cont.) CPU Usage

Expression	
	- all: sum(rate(container_cpu_usage_seconds_total{pod=~".*", namespace="\$namespace"}[2m])) - cncc-core-cmservice: sum(rate(container_cpu_usage_seconds_total{pod=~".*core-cmservice.*", namespace="\$namespace"}[2m])) - cncc-mcore-ingress: sum(rate(container_cpu_usage_seconds_total{pod=~".*mcore-ingress.*", namespace="\$namespace"}[2m])) - cncc-acore-ingress: sum(rate(container_cpu_usage_seconds_total{pod=~".*acore-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]))

8.4.2 Memory Usage

Table 8-14 Memory Usage

Description	Memory usage by all as well as individual microservices in CNCC
Expression	- all: sum (container_memory_usage_bytes{container!="", pod=~".*", namespace="\$namespace"}) - cncc-core-cmservice: sum (container_memory_usage_bytes{container!="", pod=~".*core-cmservice.*", namespace="\$namespace"}) - cncc-mcore-ingress: sum (container_memory_usage_bytes{container!="", pod=~".*mcore-ingress.*", namespace="\$namespace"}) - cncc-acore-ingress: sum (container_memory_usage_bytes{container!="", pod=~".*acore-ingress.*", namespace="\$namespace"}) - cncc-iam-ingress: sum (container_memory_usage_bytes{container!="", pod=~".*iam-ingress.*", namespace="\$namespace"}) - cncc-iam-kc: sum (container_memory_usage_bytes{container!="", pod=~".*iam-kc.*", namespace="\$namespace"})

9

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](#)

9.1 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_milliseconds>,
  "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 9-1 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 9-1 (Cont.) Log Details

Name	Description	Example
loggerName	Name of the class that generated the log	"loggerName": "ocpm.cne.gateway.cncc.GatewayApplication"
logTimestamp	Time represented in human readable format and in UTC. Format is <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 9-2 CNCC Message Format

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

Table 9-2 (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 9-2 (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

9.2 Types of Logs

The CNC Console logs can be categorized into following types:

- [Regular logs](#)
- [Audit logs](#)
- [Security logs](#)

Regular logs

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

Example:

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

```
refused (Connection  
  refused)"}}
```

Audit Logs

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

The following events are logged in CNC Console Core:

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

Note

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

The following events are logged in CNCC IAM:

Login events

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

Account events

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

Note

The user can find the CNCC IAM User Activity logs as part of *cncc-iam-0*, represented in Keycloak format. These events are provided by keycloak and documented under [Keycloak Auditing End Events](#).

Logging Error Logs are recorded by keycloak container as :

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

Security Logs

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

Disabling Security Logs

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

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

Header Information

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

Note

The user can find the Security logs :

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

Log Levels

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

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

Default log levels set for M-CNCC IAM:

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

Updating M-CNCC IAM KC log level

- By default the log level of M-CNCC IAM KC is set to **WARN,org.keycloak.events:DEBUG**
 - This means the *root log-level* for is set to **WARN** and the *org.keycloak.events* package is set to **DEBUG**
- In the **level** label set the log level, follow the below instruction about the options available to set log level
 - **Log level**
 - * TRACE
 - * DEBUG
 - * INFO
 - * WARN
 - * ERROR
 - * FATAL
 - **level: INFO**
INFO, WARN, ERROR, FATAL will be covered for logs of all packages in KC
 - **level: INFO, DEBUG**
In this case it will take the last one as log level, here DEBUG
 - **level: INFO, package-group: WARN**
What does this mean ?

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

level: INFO,org.keycloak:WARN,org.infinispan:WARN,org.jgroups:TRACE
 - **level: OFF**
No logs will appear

- **level: ALL**
logs of all the level will appear in the logs
- Sample M-CNCC IAM KC log configuration:

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

Supported Headers for Logging

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

Role of supporting headers in CNCC Audit and Security logs

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

Note

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

9.3 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=0AUTH
```

- 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=0AUTH
```

- 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=0AUTH
```

- 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=0AUTH
```

- 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=0AUTH
```

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,
```

```

        ipAddress=192.168.219.64,
        auth_method=openid-connect,
        10.75.225.28:31373/cncc/auth/admin/master/console/, code_id=5aca4960-
        eecf-406b-a7eb-92e249c2beeb,
        a7eb-92e249c2beeb,
        userId=d7cde46f-15e1-4ff8-a2cb-c5825e481438,
        error=invalid_user_credentials,
        auth_type=code, redirect_uri=http://
        username=admin,
        authSessionParentId=5aca4960-eecf-406b-
        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,
        ipAddress=192.168.219.64,
        redirect_uri=http://10.75.225.28:31373/cncc/auth/admin/master/console/,
        consent=no_consent_required,
        a7eb-92e249c2beeb, username=admin,
        a7eb-92e249c2beeb,
        code_id=5aca4960-eecf-406b-
        authSessionParentId=5aca4960-eecf-406b-
        authSessionTabId=8sruELA1Wws

```

- User created

```

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

```

- Deleted user

```

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

```

- 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","canaryTraffic":5},
{"serviceName":"namf-
comm","canaryReleaseFlag":true,"apiFullVersion":"2.0.0","canaryTraffic":5},
{"serviceName":"namf-
evts","canaryReleaseFlag":true,"apiFullVersion":"2.0.0","canaryTraffic":5},
{"serviceName":"namf-
loc","canaryReleaseFlag":true,"apiFullVersion":"2.0.0","canaryTraffic":5},
{"serviceName":"namf-
mt","canaryReleaseFlag":true,"apiFullVersion":"2.0.0","canaryTraffic":5},
{"serviceName":"nausf-
auth","canaryReleaseFlag":true,"apiFullVersion":"2.0.0","canaryTraffic":5},
{"serviceName":"nausf-
sorprotection","canaryReleaseFlag":true,"apiFullVersion":"2.0.0","canaryTra
ffic":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
```

9.4 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 9-1 Log View

