

# Oracle® Communications

## Cloud Native Core Console User's Guide



Release 1.3.0

F35044-01

September 2020

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2020, 2020, 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 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" or "commercial computer software documentation" 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 and Java 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

<b>1</b>	<b>Introduction</b>	
	Reference	1-1
	Acronyms	1-2
<b>2</b>	<b>CNC Console</b>	
	Login to CNC Console	2-1
	Working on CNC Console	2-2
	Types of User Interface Screens for NF Configurations	2-3
<b>3</b>	<b>Configuring Network Functions</b>	
	Network Repository Functions (NRF)	3-1
	Policy	3-2
	Service Communication Proxy (SCP)	3-4
	Unified Data Repository (UDR)	3-4
	Network Functions and Versions	3-6
<b>4</b>	<b>Setting up CNC Console IAM</b>	
	Setting up the CNCC Redirection URL	4-1
	Viewing the Roles in CNC Console IAM	4-3
	Users and Roles in CNC Console IAM	4-4
	Creating the Users	4-4
	Viewing the Users	4-6
	Assigning the Roles to User	4-7
	Integrating SAML SSO with CNC Console IAM	4-7
	Integrating CNC Console LDAP Server with CNC Console IAM	4-12
	Setting up User Federation with CNC Console IAM (LDAP Server integration)	4-14
	Group LDAP Mapper and Role Assignment	4-18

5	Accessing NF Resources through Curl or Postman	
6	CNC Console Metrics	
7	CNCC Logs	
	Log Formats	7-1
	Types of Logs	7-4
	Configuring Security Logs	7-6
	Examples of Logs	7-7
	Accessing logs	7-12
A	CNC Console Roles	
	Types of Roles in CNC Console	A-1
	How to Set or Update User Password in CNCC IAM	A-2
	How to Set or Update Admin Password in CNCC IAM	A-3

# 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.

---

# What's New in This Guide

This section introduces the documentation updates for Release 1.3.0 in Cloud Native Core Console.

- **New and Updated features in Release 1.3.0:**
  - CNCC Ingress Gateway Metrics
  - Helm test

# 1

## Introduction

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

In this release, CNC Console GUI provides user interface for the configuring the following Network Functions (NFs):

- Network Repository Function (NRF)
- CNC Policy
- Service Communication Proxy (SCP)
- Unified Data Repository (UDR)

This document gives a brief idea about configuring NRF, Policy, SCP and UDR network functions in CNC Console GUI.

The user can edit, update or delete the parameters of these NFs.

The **Setting up CNC Console IAM** section describes the authentication and authorization. It describes how an **Administrator** can:

- Setup the redirection URL
- View Roles in CNC Console IAM
- Create Users
- View the Users
- Assign Roles to User
- SAML SSO integration
- LDAP Server integration in CNC Console IAM
- Access NF Resources through curl or postman

### Note:

Currently CNC Console supports only within cluster deployment.

## Reference

Refer the following documents for more information:

- Service Communication Proxy (SCP) Cloud Native User's Guide
- Network Repository Function (NRF) Cloud Native User's Guide
- Cloud Native Core Policy User's Guide
- Unified Data Repository (UDR) Cloud Native User's Guide

- Network Repository Function (NRF) Cloud Native Installation and Upgrade Guide
- Service Communication Proxy (SCP) Cloud Native Installation Guide
- Unified Data Repository (UDR) Cloud Native Installation and Upgrade Guide
- Cloud Native Core Policy Installation Guide

## Acronyms

**Table 1-1 Acronyms**

Terms	Definition
CNCC	Cloud Native Core Console
HTTPS	Hypertext Transfer Protocol Secure
IAM	Identity Access Management
LDAP	Lightweight Directory Access Protocol
NRF	Network Repository Function
OSDC	Oracle Software Delivery Cloud
SCP	Service Communication Proxy
SAML	Security Assertion Markup Language
UDR	Unified Data Repository
UE	User Equipment



# 2

## CNC Console

This section provides information about CNC Console.

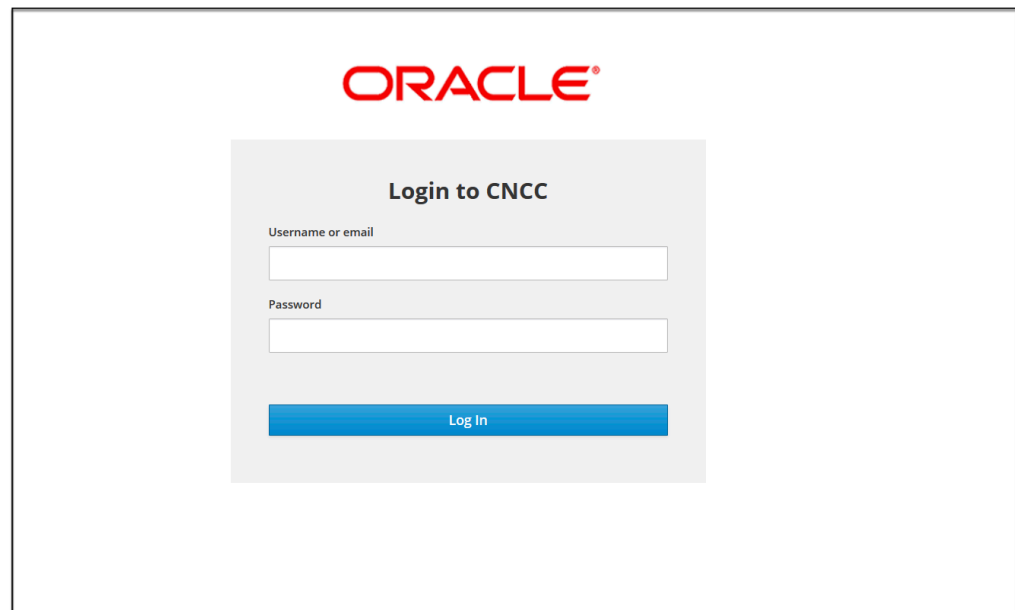
The CNC Console has two modules:

- **CNC Console Core (CNCC Core):** CNCC Core module includes the GUI aspects of the interface. The integration of all the supported NFs are included in this module.
- **CNC Identity Access Management (CNC IAM):** CNC IAM module includes the authentication and authorization aspects of the interface. This includes creating and assigning roles to users.

### Login to CNC Console

The procedure to login to the CNC Console is as follows:

1. Open any browser.
2. Enter the URL: **http://<host name>:<port number>**. The **Log In** screen appears:



 **Note:**

*<host name> is cnc-iam-ingress-ip and <port number> is cnc-iam-ingress-port*

3. Enter the valid credentials.

4. Click **Log In**. The Welcome Screen of CNC Console interface appears.



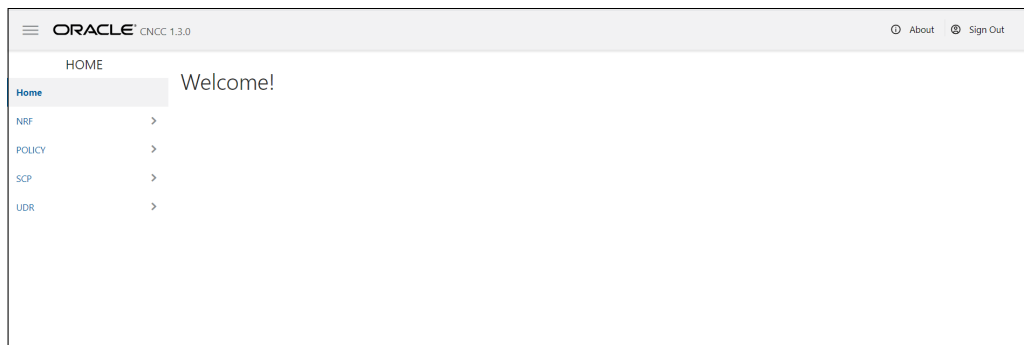
**Note:**

To set up CNCC IAM, refer to [Setting up CNC Console IAM](#) section

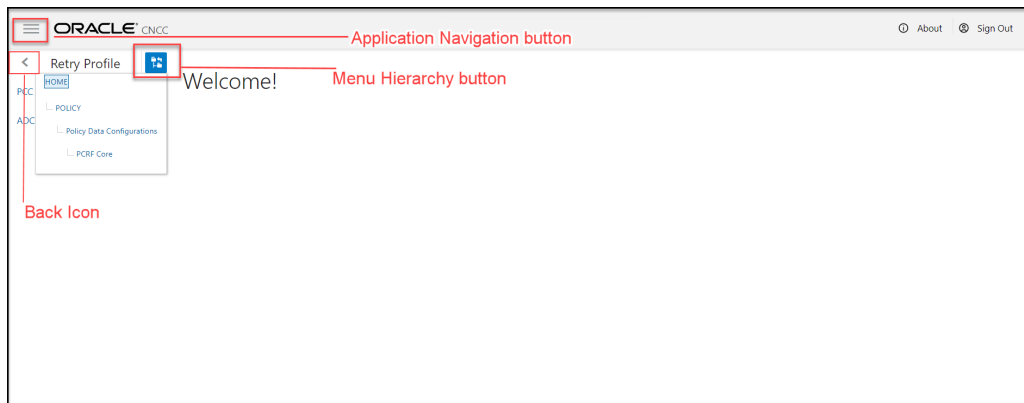
## Working on CNC Console

### GUI Details

After the user log in using credentials, the CNC Console Welcome screen appears by default.



- 1. Top Ribbon** -The top ribbon has following features:
  - **About**- Tells about the product name and the version of the Interface.
  - **Sign Out**- To sign out from the Console.
- 2. Left Pane - NFs and APIs**  
The left pane displays the list of Network Functions and respective configurations.
- 3. Right Pane - Details View**  
The right pane displays the configurable parameters that can be updated in the selected NFs .



 **Note:**

- The **Menu Hierarchy** button shows the navigation path 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 navigates to the Home menu. Screen does not get refreshed automatically. User must click Home/NF menu to view the updated screen.

## Types of User Interface Screens for NF Configurations

### CNC Console Screens

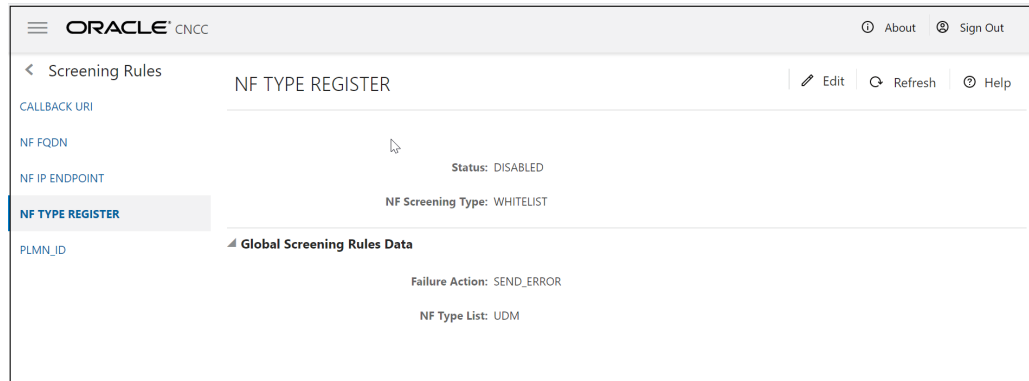
CNC Console has two types of screens:

#### Service Screen

Service Screens has single independent objects. These kinds of screen are used to display and configure single object.

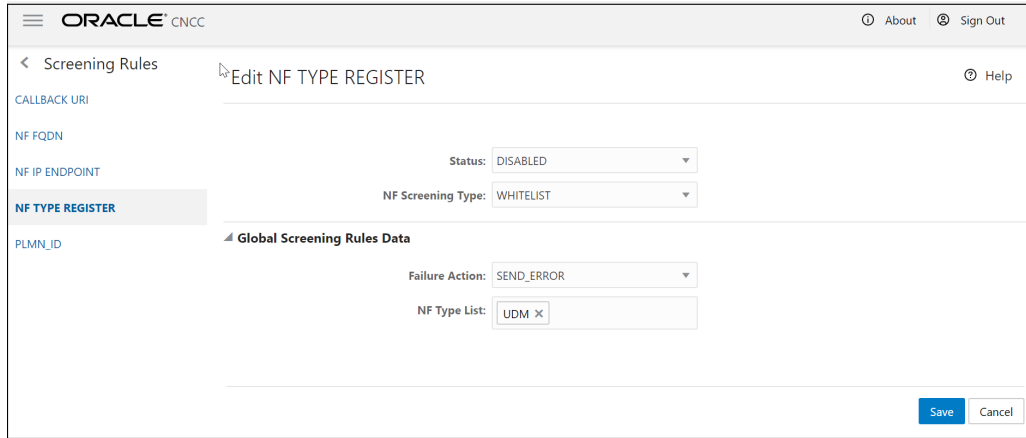
Examples for the Service Screen:

#### Service Screen



#### Service Screen - Edit

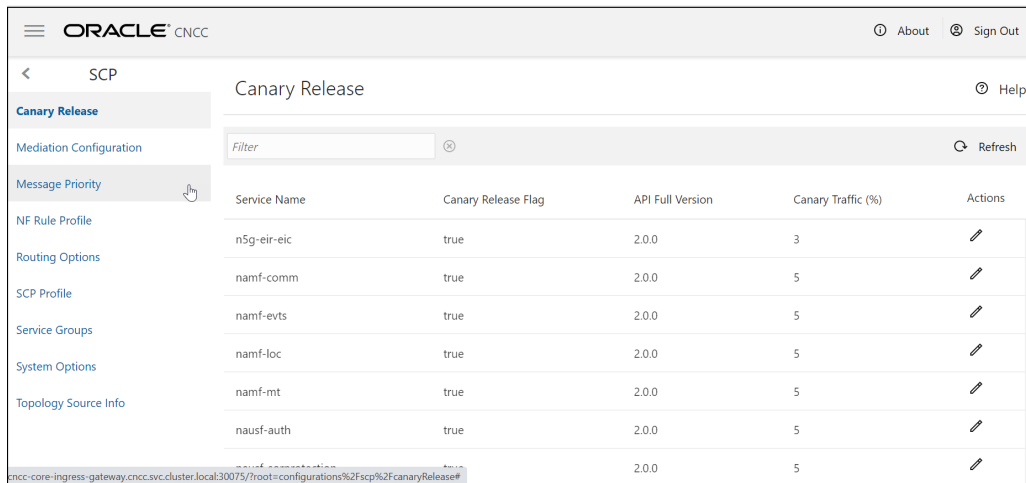
The Service screen with **Edit** enabled. The user can update the parameters.



### Configurations Screen

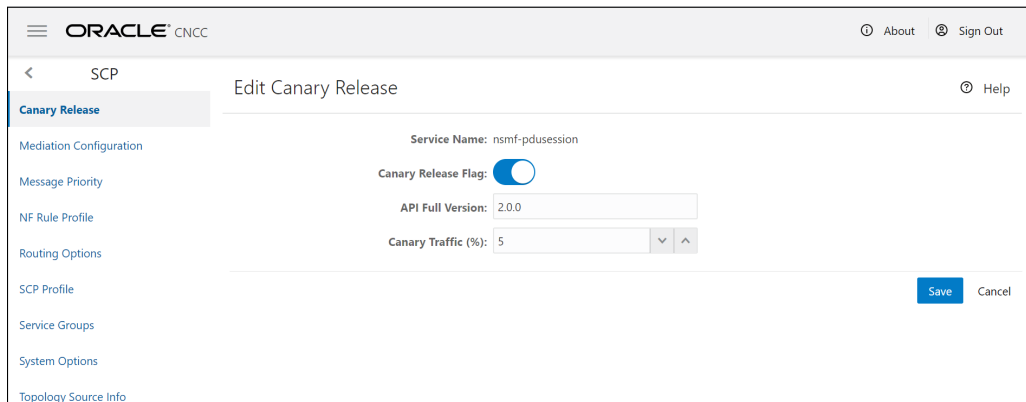
The Configurations screen is used to display and configure multiple related objects.

Examples for the Configuration Screen:



### Configurations Screen- Edit

The Configurations screen with **Edit** enabled. The user can update the parameters.



## Configurations Screen - Add

The Configurations screen with **Add** enabled.

The screenshot shows the Oracle CNCC user interface for adding a mediation configuration. The top navigation bar includes the Oracle logo and 'CNCC', along with 'About' and 'Sign Out' links. The breadcrumb trail shows 'SCP' and the page title is 'Add Mediation Configuration'. A sidebar on the left lists various configuration categories, with 'Mediation Configuration' selected. The main content area contains several input fields: 'Group ID', 'Message Type', 'NF Service', 'NF Type' (a dropdown menu), and 'Mediation Trigger Points'. Below these fields is a section titled 'Match' containing a table with a header 'Name' and an 'Add' button. The table is currently empty, displaying the message 'No data to display.' At the bottom right of the screen, there are 'Save' and 'Cancel' buttons.

# 3

## Configuring Network Functions

### Network Repository Functions (NRF)

#### Overview

The NRF is a key component of the 5G Service Based Architecture. The NRF maintains an updated repository of all the Network Functions (NFs) available in the operator's network along with the services provided by each of the NFs in the 5G core that are expected to be instantiated, scaled and terminated with minimal to no manual intervention. In addition to serving as a repository of the services, the NRF also supports discovery mechanisms that allows NFs to discover each other and get updated status of the desired NFs.

#### Configuring NRF Parameters

On selecting NRF on the left navigation pane the following screen appears:



1. On selecting **Screening Rules**, the functionalities of **Screening Rules** appear underneath. The functionalities are **CALLBACK URI, NF FDQN, NF IP ENDPOINT, NF TYPE REGISTER** and **PLMN\_ID**.
2. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
3. Click **Edit** modify the parameters.
4. On selecting **System Options**, the parameters of **System Options** appear on the right pane.
5. Click **Edit** to modify the parameters.
6. Click **Save**.

**Note:**

For details about configurable parameters, refer to **Network Repository Function (NRF) Cloud Native User's Guide**.

## Policy

### Overview

Oracle Communications Cloud Native Core Policy (CNC Policy) solution provides a standard policy design experience and ultimately consistent end-user experience. The Converged policy solution supports both 4G and 5G networks. In addition, the overlap in functionality between PCF and PCRF (Example: need for a policy engine, policy design, Rx, similarity between Sy and Nchf\_SpendingLimitControl, etc.) enables us to build micro-services that can be used to provide PCRF and PCF functionality. Even though it is a unified policy solution, you can still deploy the PCF and PCRF entirely independently.

The CNC Policy is a functional element for policy control decision and flows based charging control functionalities. The CNC Policy provides the following functions:

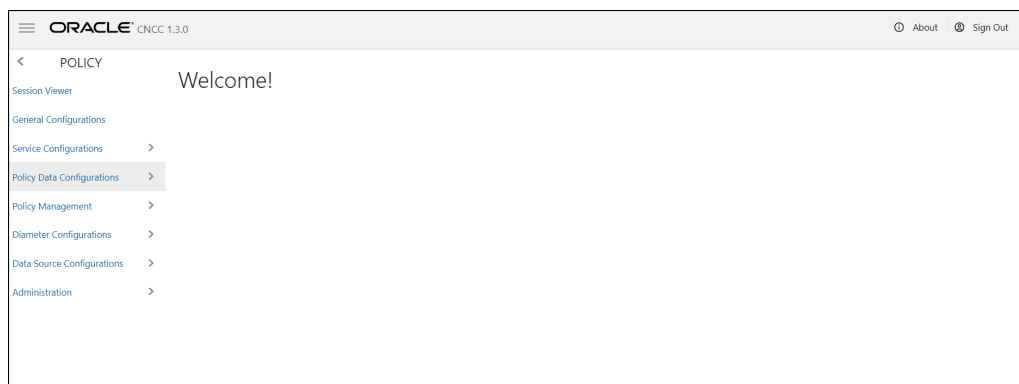
- Policy rules for application and service data flow detection, gating, QoS, and flow based charging to the Session Management Function (SMF)
- Access and Mobility Management related policies to the Access and Mobility Management Function (AMF)
- Provide UE Route Selection Policies (URSP) rules to UE via AMF
- Accesses subscription information relevant for policy decisions in a Unified Data Repository (UDR)
- Provides network control regarding the service data flow detection, gating, QoS and flow based charging towards the Policy and Charging Enforcement Function (PCEF).
- Receives session and media related information from the AF and informs AF of traffic plane events.
- Provisions PCC Rules to the PCEF via the Gx reference point.

The CNC Policy supports the above functions through the following services:

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

### Configuring Policy Parameters

On selecting **POLICY** on the left navigation pane the following screen appears:



1. On selecting **General Configurations**, the parameters of **General Configurations** appear on the right pane.
2. Click **Edit** to modify the parameters.
3. On selecting **Service Configurations**, the different PCF and PCRF services appear underneath the **Service Configurations**. Different services are: **PCF Session Management, PCF Access and Mobility, PCF Policy Authorization, PCF UE Policy, PCF User Connector, PCRF Core, Audit and Policy Engine**.
4. On selecting a service, the configurable parameters of the functionality appear on the right pane.
5. Click **Edit** to modify the parameters.
6. On selecting **Policy Data Configurations**, the policy types of **Policy Data Configurations** appear underneath. The policy types are **Common, PCF Session Management, PCF Access and Mobility, PCF UE Policy and PCRF Core**.
7. On selecting a policy type, the respective functionalities appear underneath, and on selecting a functionality the configurable parameters of the functionality appear on the right pane.
8. Click **Edit** to modify the parameters.
9. On selecting **Policy Management**, the functionalities of **Policy Management** appear underneath. The functionalities are **Policy Projects, Policy Library and Policy Tests**.
10. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
11. Click **Edit** to modify the parameters and click **Save**.
12. On selecting **Diameter Configurations**, the **Setting, Peer Nodes and Routing Table** appear underneath.
13. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
14. Click **Edit** to modify the parameters and click **Save**.
15. On selecting **Data Source Configurations**, the **Data Sources** appear underneath.
16. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
17. Click **Add** to add the parameters and click **Save**.



- On selecting **Administration**, the **Import** and **Export** options appear underneath. On selecting an option, user can import or export the configurations.



**Note:**

For details refer to **Cloud Native Core Policy User's Guide**.

## Service Communication Proxy (SCP)

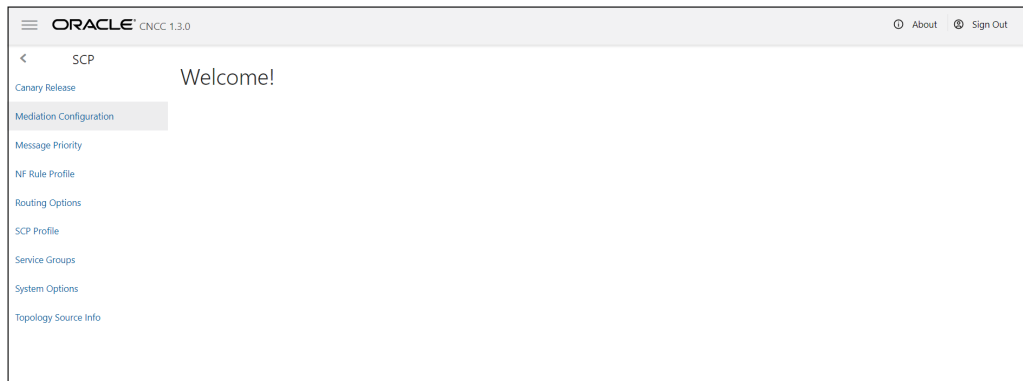
### Overview

This section provides steps to update the various configurations parameters of different APIs supported by SCP NF.

The SCP is a decentralized solution and composed of Service Proxy Controllers and Service Proxy Workers and is deployed along side of 5G network functions and provides routing control, resiliency, and observability to the core network.

### Configuring SCP Parameters

On selecting **SCP** on the left navigation pane the following screen appears:



- The options appear underneath the **SCP** are **Canary Release, Mediation Configuration, Message Priority, NF Rule Profile, Routing Options, SCP Profile, Service Groups, System Options** and **Topology Source Info**.
- On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
- Click **Edit** to modify the parameters.
- Click **Save**.



**Note:**

For details refer to **Service Communication Proxy (SCP) Cloud Native User's Guide**.

## Unified Data Repository (UDR)

UDR is a converged repository, which is used by 5G Network Functions to store the data.

Oracle 5G UDR is implemented as cloud native function and it offers a unified database for storing application, subscription, authentication, service authorization, policy data, session binding and application state information. It exposes a HTTP2 based RESTful API for Network Functions and provisioning clients to access the stored data.

Oracle's 5G UDR:

- Leverages a common Oracle Communications Cloud Native Framework.
- Is compliant to 3GPP Release 15 specification for PCF and UDM.
- Has tiered architecture providing separation between the connectivity, business logic and data layers.
- Uses Oracle MySQL Cluster CGE database technology for backend database in the DB tier.
- Registers with NRF in the 5G network, so the other NFs in the network can discover UDR through NRF.

### Configuring UDR Parameters

On selecting **UDR** on the left navigation pane the following screen appears:



1. The options appear underneath the **UDR** are **Global Configurations**, **Service Configurations** and **Provisioning**.
2. On selecting **Global Configurations** the parameters of **Global Configurations** appear on the right pane.
3. Click **Edit** to modify the parameters.
4. On selecting **Service Configurations**, the functionalities of **Service Configurations** appear underneath. The functionalities are **Data Repository Service**, **Notify Service** and **NRF Client Service**.
5. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
6. Click **Edit** to modify the parameters and click **Save**.
7. On selecting **Provisioning**, the functionalities of **Provisioning** appear underneath. The functionalities are **Profile Data**, **PCF Data**, **SLF Data**, **UDM Data** and **Schema Management Data**.
8. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
9. Click **Edit** to modify the parameters and click **Save**.



**Note:**

For details about configurable parameters, refer to **Unified Data Repository (UDR) Cloud Native User's Guide**.

## Network Functions and Versions

Following are the supported NF versions for CNC Console 1.3.0:

NF	NF Version
NRF	1.8.0
Policy	1.8.0
SCP	1.8.0
UDR	1.8.0

# 4

## Setting up CNC Console IAM

The **Administrator** can access and set up the CNC Console IAM configurations for:

- [Setting the CNCC Redirection URL](#)
- [Viewing the Roles in CNC Console IAM](#)
- [Users and Roles in CNC Console IAM](#)
  - [Creating the Users](#)
  - [Viewing the Users](#)
  - [Assigning Roles to Users](#)
- [Integrating SAML SSO with CNC Console IAM](#)
- [Integrating LDAP Server in CNC Console IAM](#)
- [Setting up User Federation with CNC Console IAM \(LDAP Server integration\)](#)
- [Group LDAP Mapper and Role Assignment](#)
- [Accessing NF Resources through curl or postman](#)

### Setting up the CNCC Redirection URL

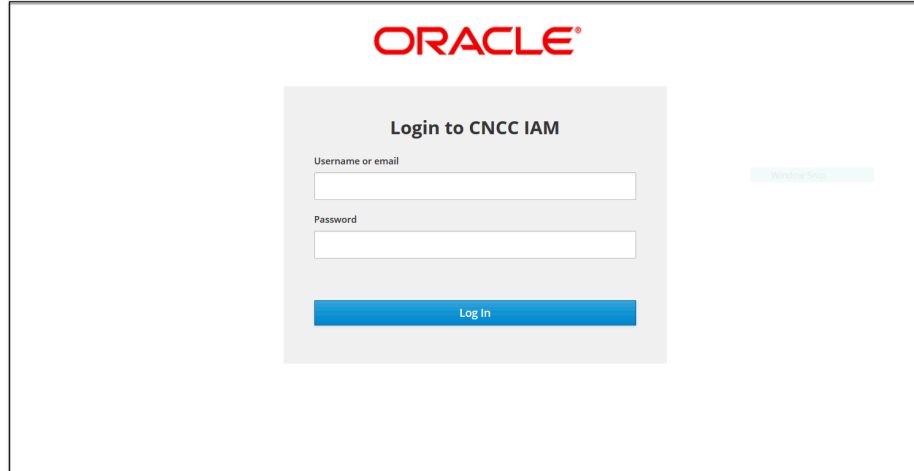
Once CNCC IAM is deployed administrator must do the setting of cncc redirection URL:

**To set up the cncc redirection URL:**

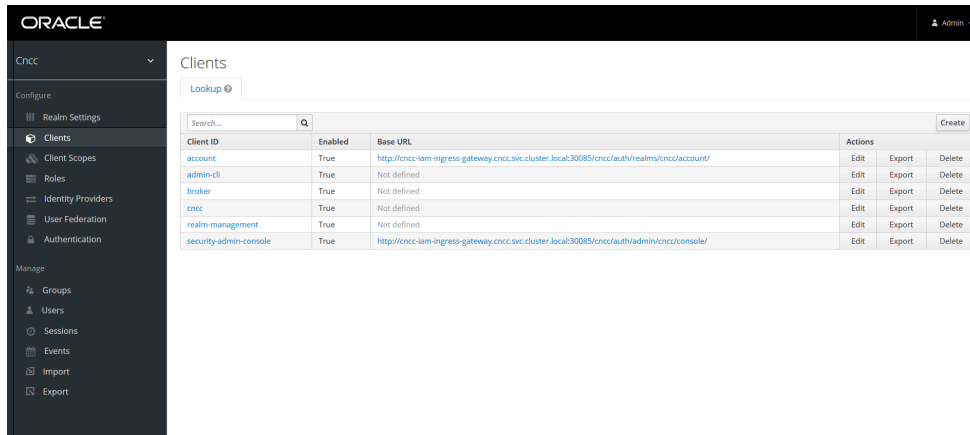
1. Login to CNCC IAM Console using admin credentials provided during installation of CNCC IAM.

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

Example: `http://10.75.182.72:8080/*`

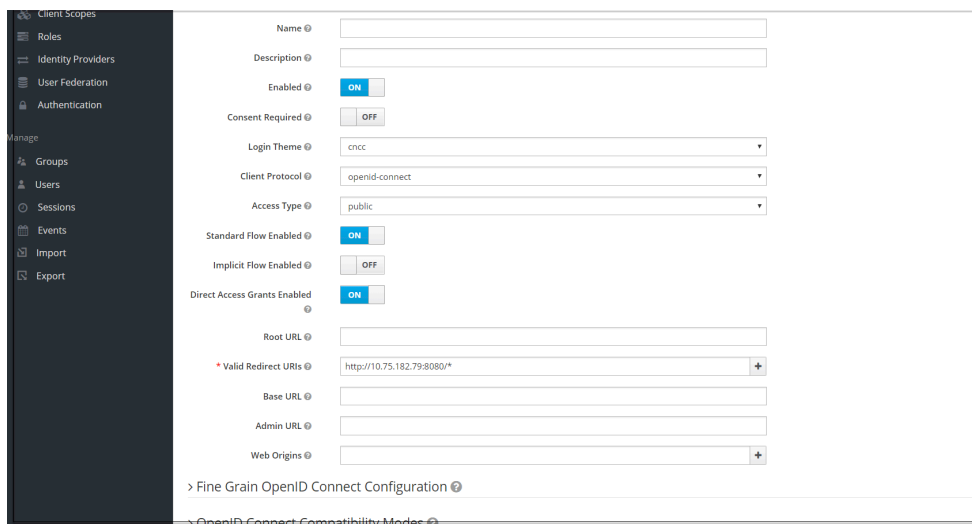


2. Go to Clients and select Cncc.



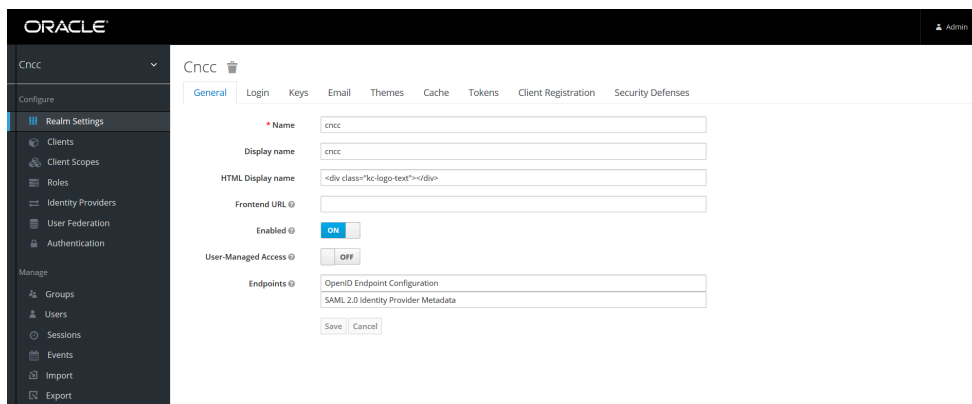
3. Enter CNCC Core Ingress URI in the **Valid Redirect URIs** field and click **Save**.

<scheme>://<cncc-core-ingress-extrenal-ip>:<cncc-core-ingress-service-port>/\*  
Example: http://10.75.182.79:8080/\*

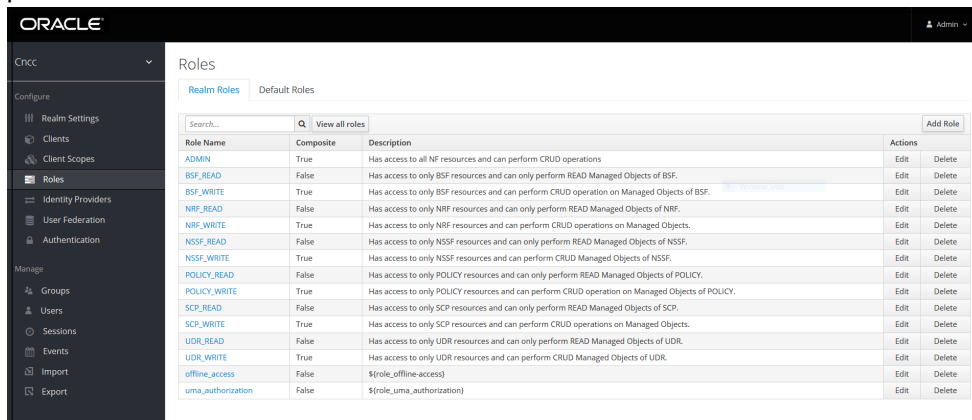


## Viewing the Roles in CNC Console IAM

1. Select **Realm Settings** under **Cnc**.



2. Click **Roles** in the left pane. The roles defined in that realm appears in the right pane.



# Users and Roles in CNC Console IAM

This section includes:

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



### Note:

For more details about the user roles refer [APPENDIX](#).



### Note:

For the details about Setting or updating the admin password refer [How to Set or Update Admin Password in CNCC IAM](#)

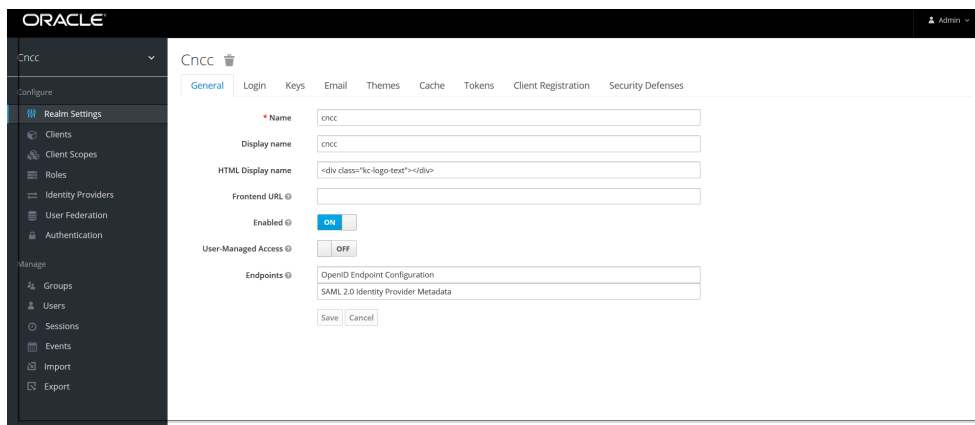


### Note:

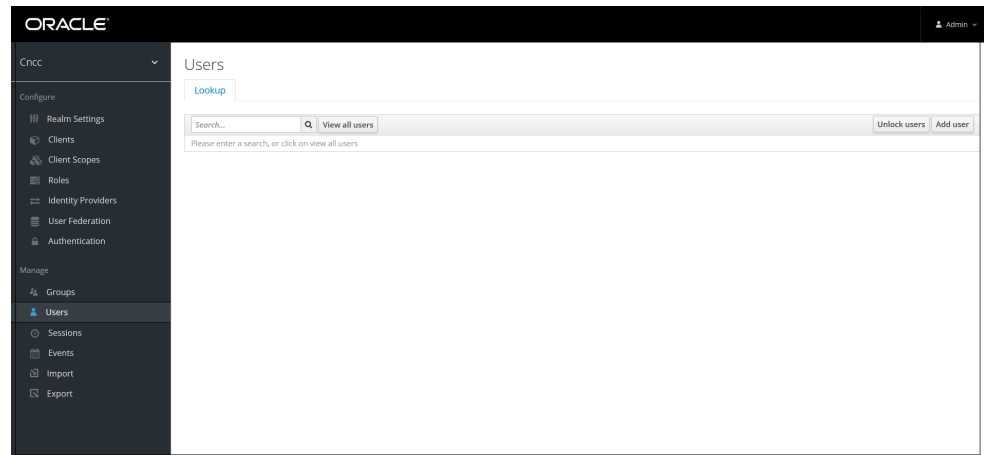
For the details about Setting or updating the user password refer [How to Set or Update User Password in CNCC IAM](#)

## Creating the Users

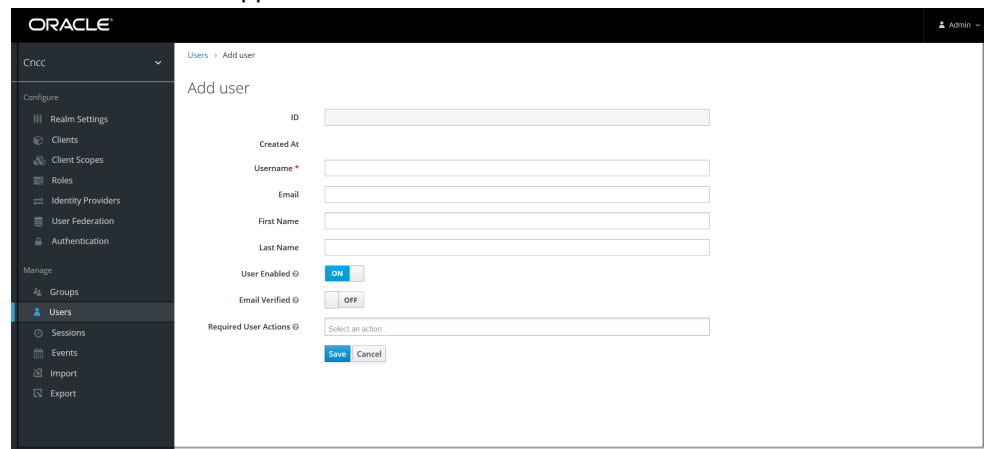
1. Click **Realm Settings** under **Cncc**.



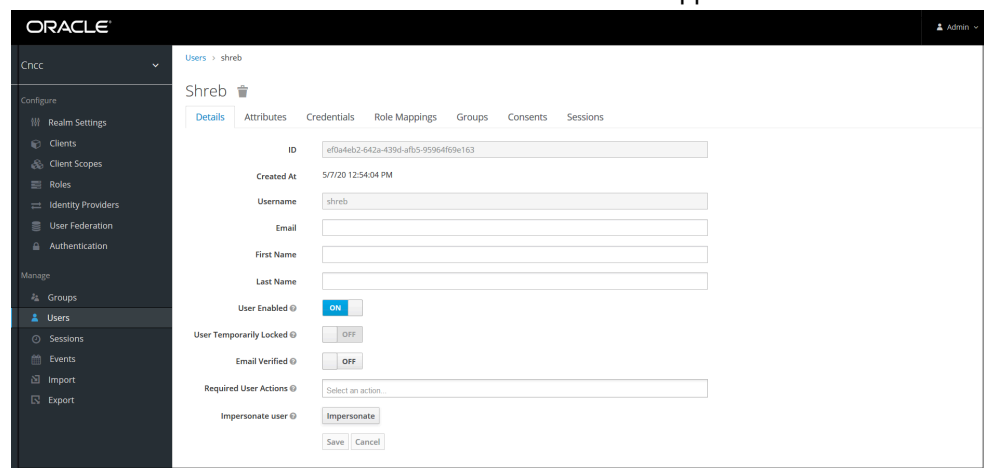
2. Select **Users** under **Manage** in the left pane and select **Add user** in the right pane.



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

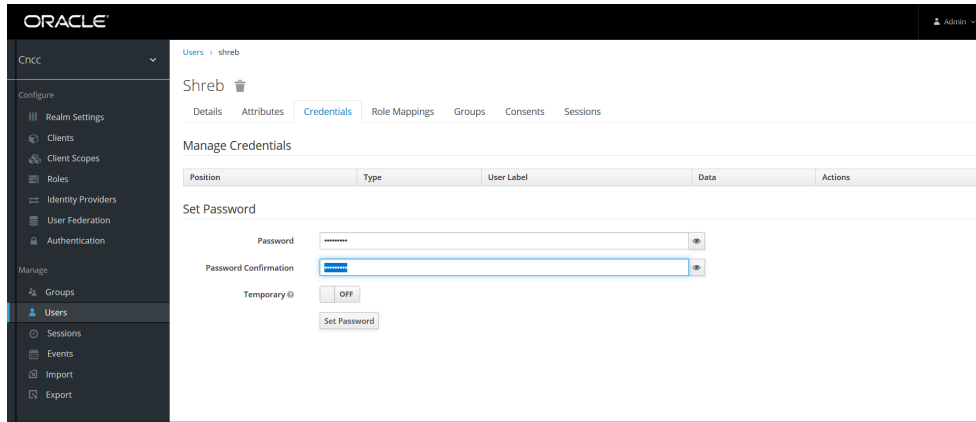


4. The user has been created and the user details screen appears.



5. For setting the password for the user, Select **Users** under **Manage** in the left pane and select the **Credentials** tab in the right pane. Set the password for the user.



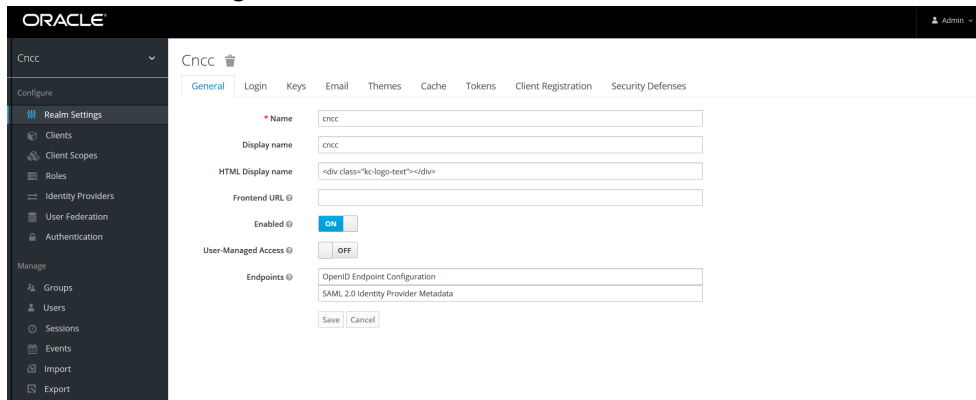


**Note:**

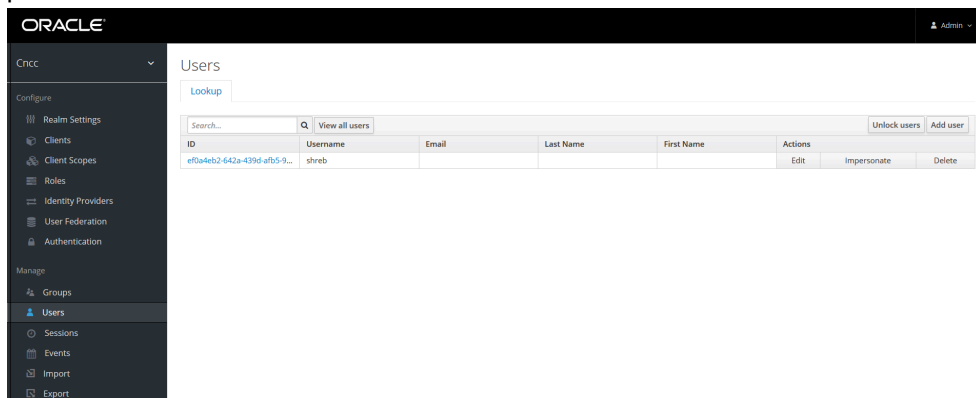
Setting the **Temporary** flag **ON** prompts the user to change the password while login for the first time to CNC Console Interface.

## Viewing the Users

1. Click **Realm settings**.



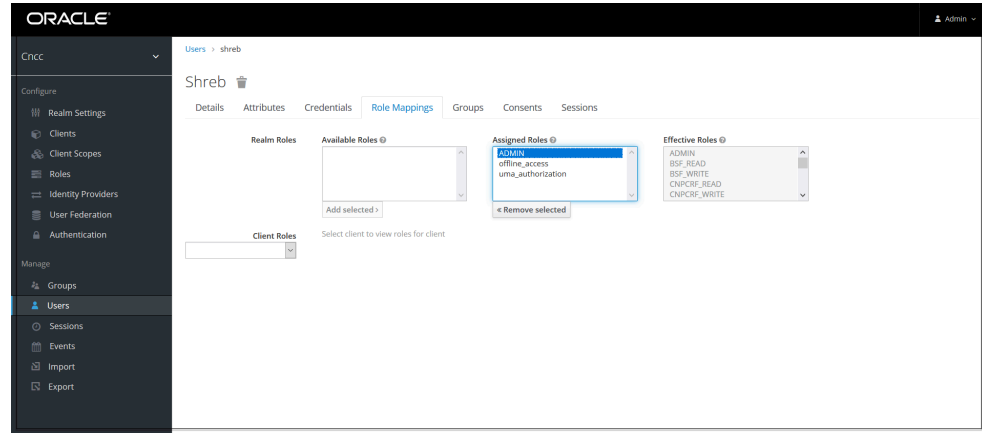
2. Select **Users** under **Manage** in the left pane and select **View all users** in the right pane.



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

## Assigning the Roles to User

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



The selected roles will be assigned to the user.

## Integrating SAML SSO with CNC Console IAM

### Overview

Security Assertion Markup Language (SAML) is an open standard that allows identity providers (IdP) to pass authorization credentials to service providers (SP). The identity provider authenticates the user and returns the assertion information about the authenticated user and the authentication event to the application. If the user tries to access any other application that uses the same identity provider for user authentication, the user shall not be required to log in a second time and will be granted access. This is the principle of SSO (Single Sign On).

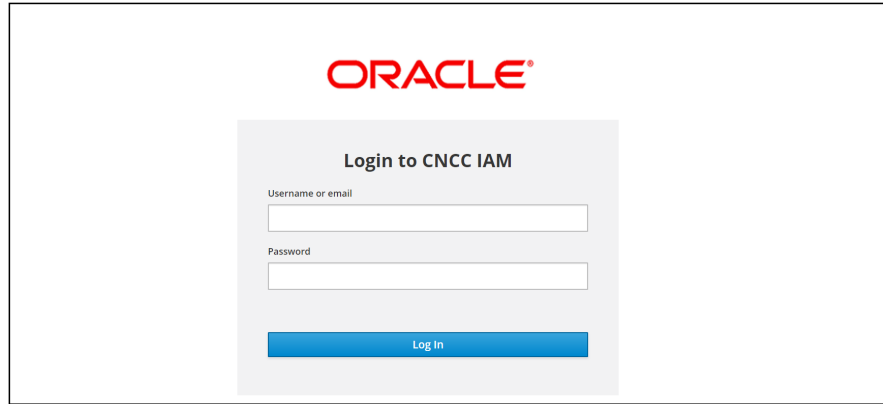
CNCC supports SAML 2.0.

### Configuring SAML Identity Provider in CNCC IAM

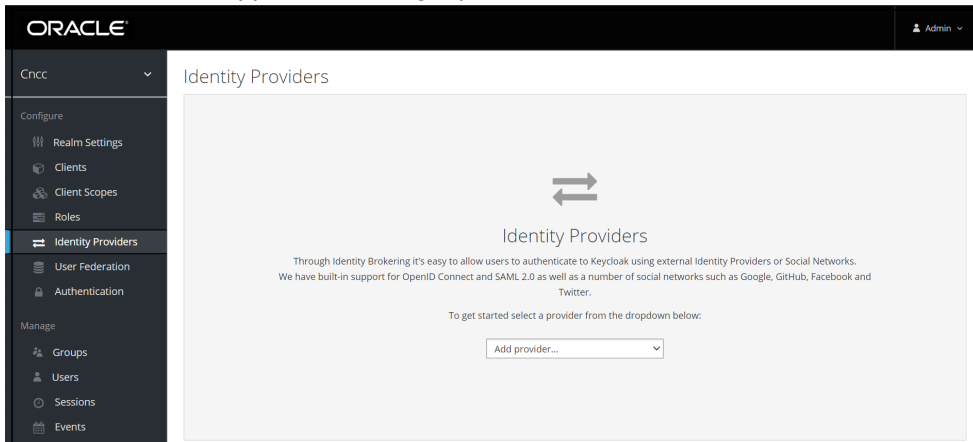
1. To configure SAML identity provider (**IdP**) in CNCC IAM, login to CNCC IAM Console using admin credentials provided during installation of CNCC IAM .

`http://<cnc-iam-ingress-extrenal-ip>:<cnc-iam-ingress-service-port>`

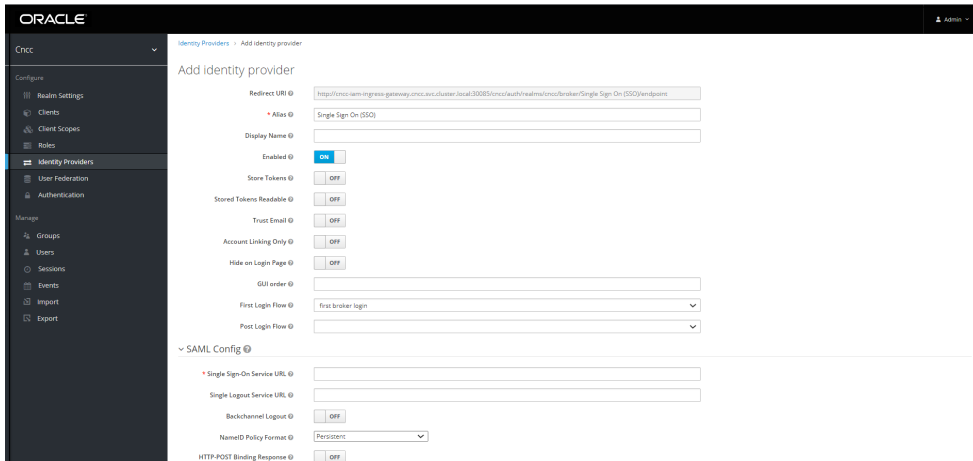
Example: `http://cnc-iam-ingress-gateway.cnc.svc.cluster.local:30085/`



2. Select **Cncc** realm and the **Identity Provider** tab in the left pane. **Identity Providers** screen appears in the right pane.



3. From the **Add provider** drop down list select the **saml** entry and the **Add Identity Provider** screen appears.

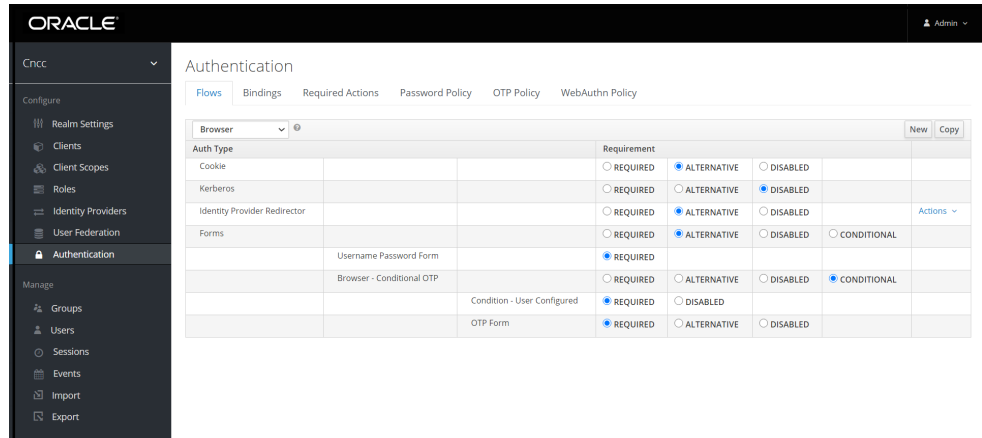


 **Note:**

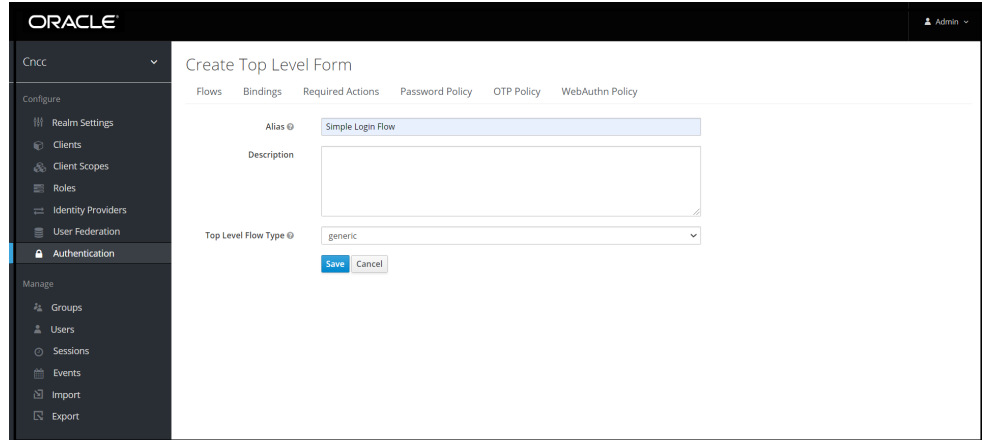
- Give an appropriate name for the field **Alias**.
- At **Import External IDP Config**, upload the 'idp-metadata.xml' file that is exported from SAML client in the IdP.

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

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

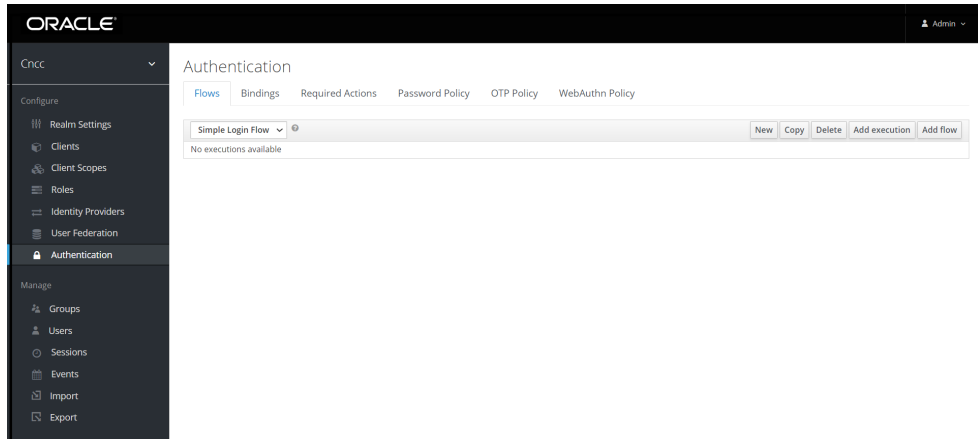


5. Click **New** at the right pane. **Create Top Level Form** screen appears.

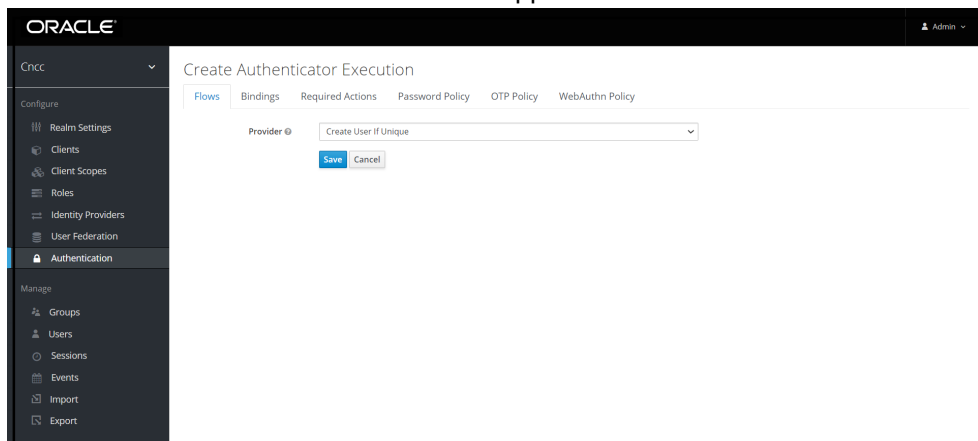


Enter the appropriate alias and click **Save**.

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

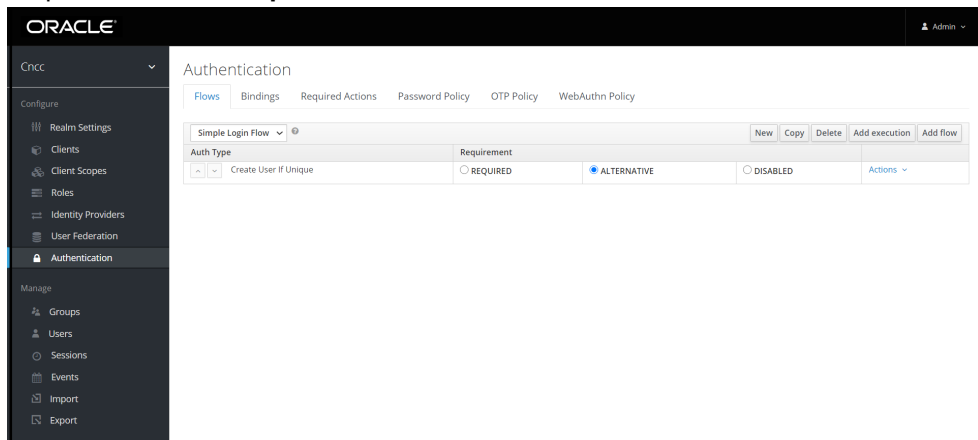


7. Create Authenticator Execution screen appears.

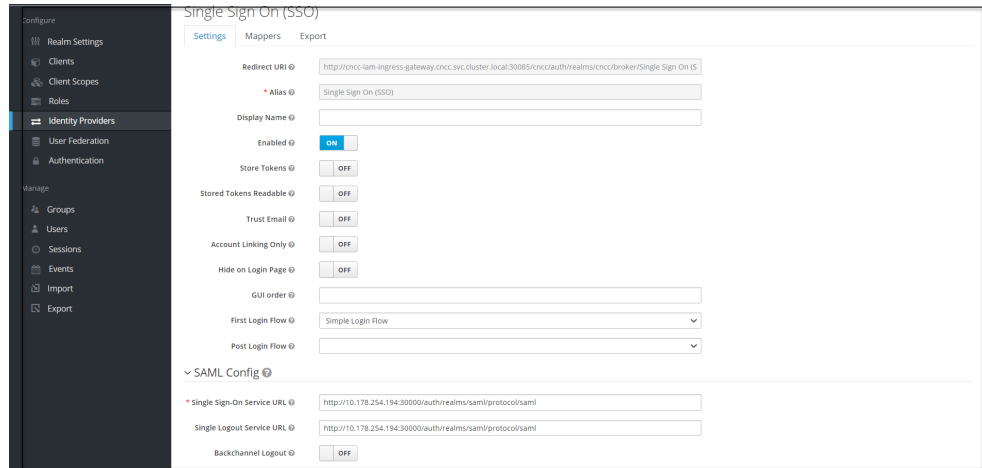


Select **Create User If Unique** from the **Provider** drop down list. Click **Save**.

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

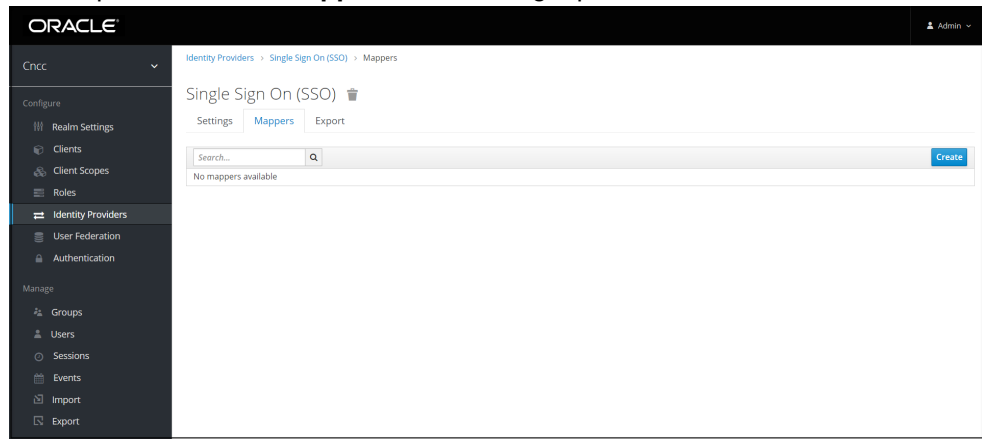


9. Select **Identity Provider** in the left pane. Select the custom flow from **First Login Flow** drop down list.

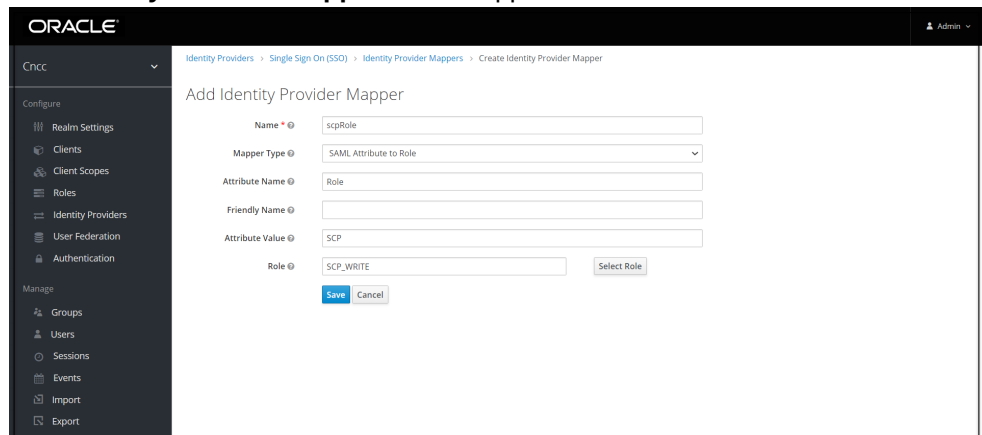


### Mapping SAML IdP roles with CNCC IAM API roles

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

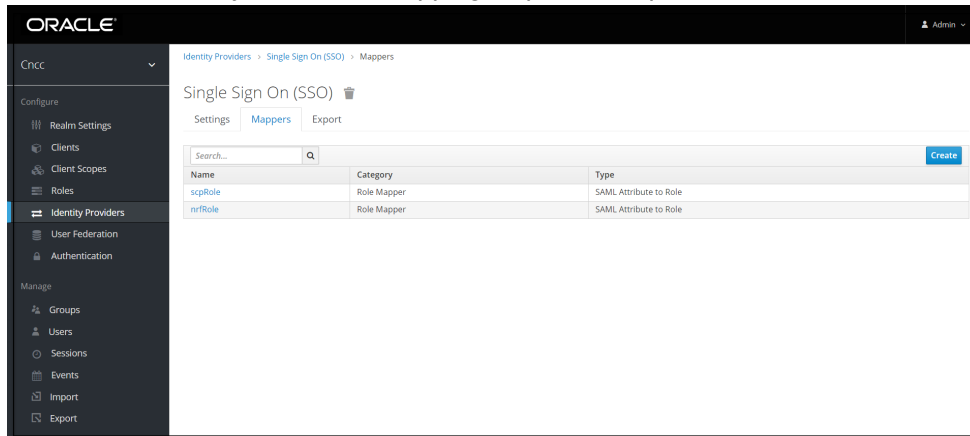


2. Add Identity Provider Mapper Screen appears.



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

- Click **Select Role** to select the API roles to be enabled for this mapping.
  - Click **Save**.
3. User can create any number of mapping as per the requirements.

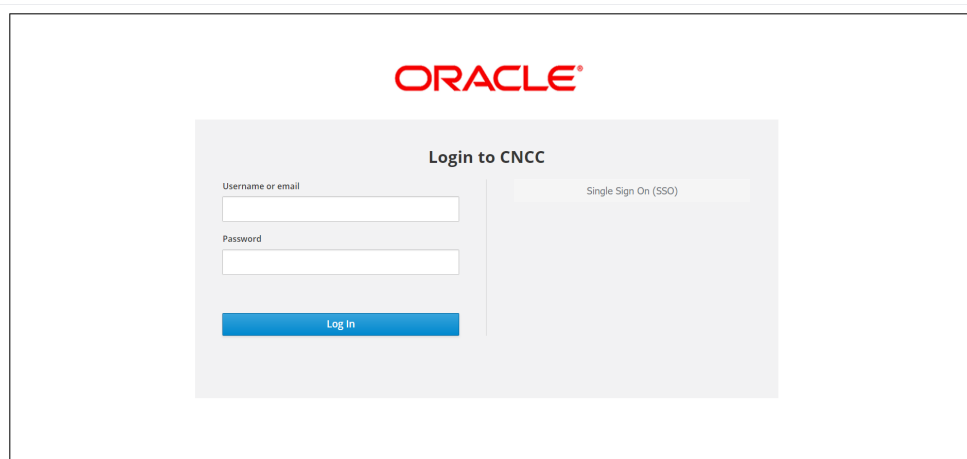


### Accessing CNCC Core Application

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

`http://<cncc-iam-ingress-external-ip>:<cncc-iam-ingress-service-port>`

Example: `http://cncc-core-ingress-gateway.cncc.svc.cluster.local:30075/`



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

## Integrating CNC Console LDAP Server with CNC Console IAM

### Overview

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

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

### 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
```



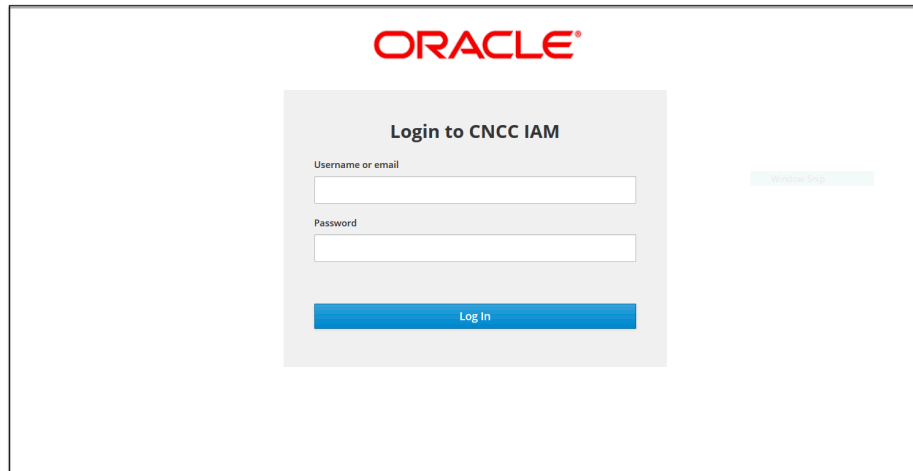
```
objectclass: groupOfUniqueNames
cn: admin
uniqueMember: uid=ben,ou=people,dc=oracle,dc=org
ou: admins

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

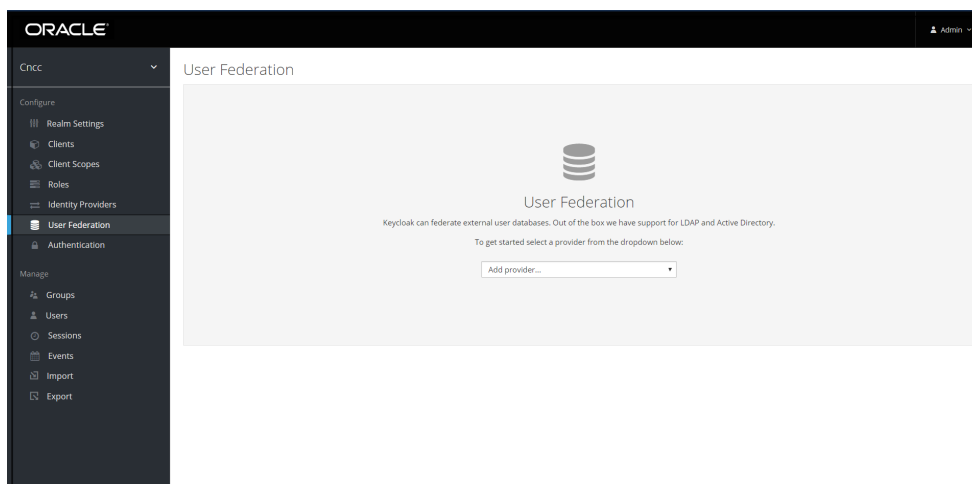
dn: cn=nrf,ou=groups,dc=oracle,dc=org
objectclass: top
objectclass: groupOfUniqueNames
cn: nrf
uniqueMember: uid=ben,ou=people,dc=oracle,dc=org
uniqueMember: uid=bob,ou=people,dc=oracle,dc=org
ou: nrfusers
```

## Setting up User Federation with CNC Console IAM (LDAP Server integration)

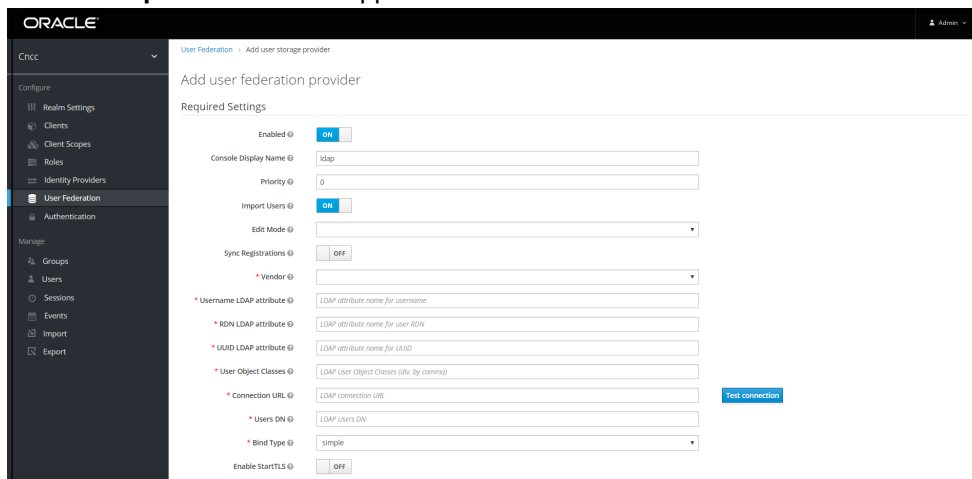
1. Go to CNCC IAM console <http://<cncc-iam-ingress-ip>:<cncc-iam-ingress-port>> and login using admin credentials provided during installation of CNCC IAM.



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



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



- Fill the following parameters:
  - Console Display Name:** Enter the display name.
  - Vendor:** Enter the LDAP server provider name for the company.

 **Note:**

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

The screenshot shows the 'Required Settings' configuration page for an LDAP provider. The left sidebar contains navigation options: Home, Clients, Client Scopes, Roles, Identity Providers, User Federation, and Authentication. The main content area is titled 'Required Settings' and includes the following fields:

- Enabled:
- Console Display Name: ldap
- Priority: 0
- Import Users:
- Edit Mode: [Dropdown]
- Sync Registrations:
- Vendor: Other
- Username LDAP attribute: uid
- RDN LDAP attribute: uid
- UID LDAP attribute: uid
- User Object Classes: inetOrgPerson, organizationalPerson, person, top
- Connection URL: [Text field]
- Users DN: ldap://users.dn
- Bind Type: simple
- Enable StartTLS:
- Bind DN: [Text field]
- Bind Credential: [Text field]
- Custom User LDAP Filter: [Text field]
- Search Scope: One Level
- Validate Password Policy:

Buttons for 'Test connection' and 'Test authentication' are visible on the right side of the form.

- Most companies have the **UUID LDAP attribute** value set as "entryUUID". If you don't have this field, than just use another unique identifier.
- The default setting for **Import Users** is 'ON'. Change it to 'OFF' to disable user sync.
- Provide company LDAP server details.
- If the LDAP is secured then select 'simple' from the **Bind Type** drop down and provide the admin bind username and password else select **Bind Type** as "none". Sample data for the field **Bind DN** "cn=admin,dc=oracle,dc=org".
- Click "Test Connection" and "Test Authentication".
- Set Cache policy as "NO\_CACHE".

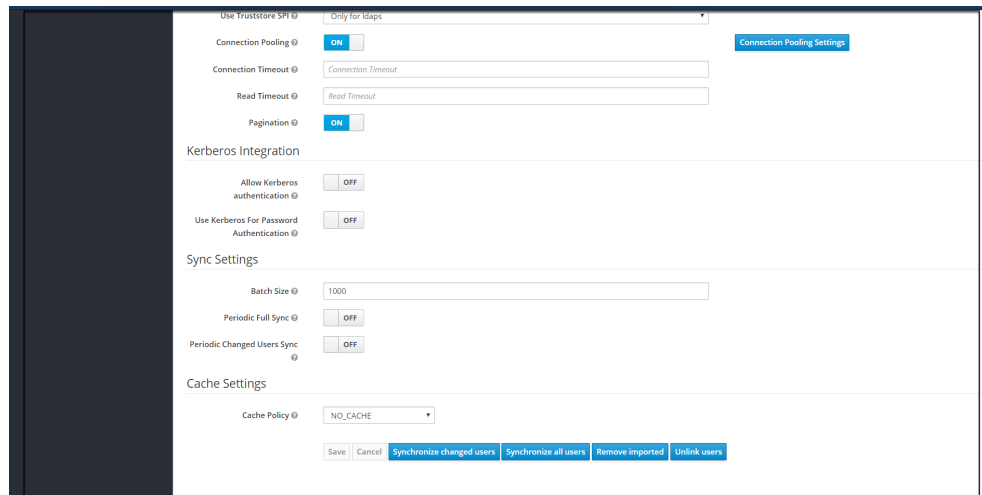
5. After filling the required fields, the screen appears as below. Click **Save**.

The screenshot shows the 'Required Settings' configuration page after saving. The configuration is as follows:

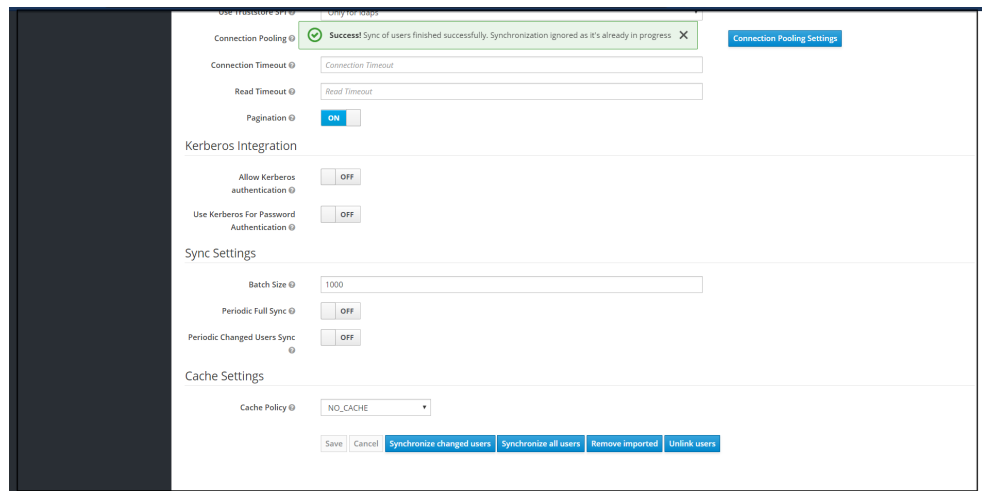
- Provider ID: c5c860ff-e5ca-4a78-934c-356c1656c705
- Enabled:
- Console Display Name: ldap
- Priority: 0
- Import Users:
- Edit Mode: [Dropdown]
- Sync Registrations:
- Vendor: Other
- Username LDAP attribute: uid
- RDN LDAP attribute: uid
- UID LDAP attribute: uid
- User Object Classes: inetOrgPerson, organizationalPerson, person, top
- Connection URL: ldap://10.178.254.194:30048
- Users DN: ou=people,dc=oracle,dc=org
- Bind Type: simple
- Enable StartTLS:
- Bind DN: cn=admin,dc=oracle,dc=org
- Bind Credential: [Text field]
- Custom User LDAP Filter: [Text field]

Buttons for 'Test connection' and 'Test authentication' are visible on the right side of the form.

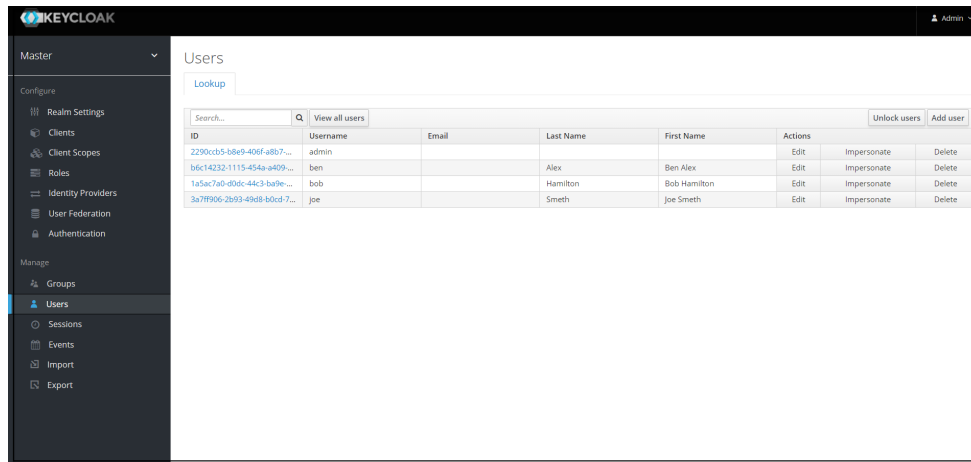
6. New buttons (**Synchronize changed users, Synchronize all users, Remove imported, Unlink users**) appears next to the **Save** and **Cancel**.



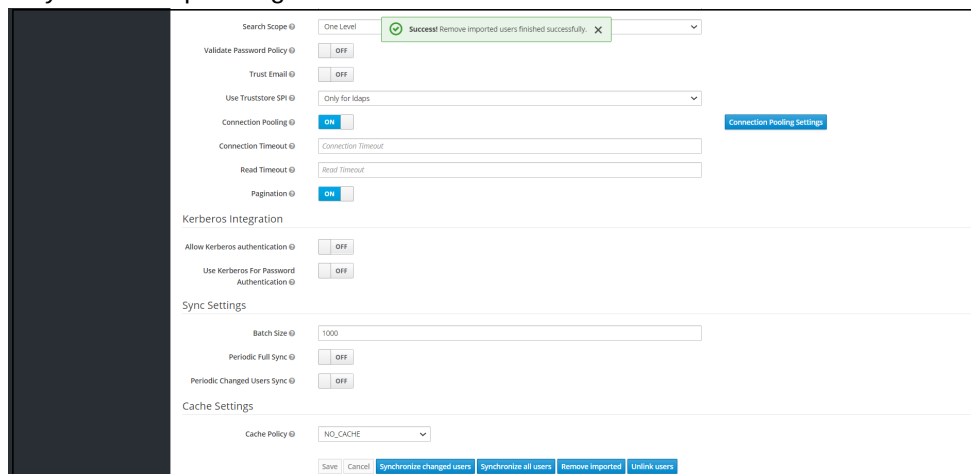
7. If a user has to be import to CNCC-IAM, Click **Synchronize all users**. If the synchronization is successful, the success message appears. If the synchronization fails, then check the trouble shooting section and look at **cncc-iam logs** in debug mode.



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



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



### Note:

The steps 8 and 9 are optional.

## Group LDAP Mapper and Role Assignment

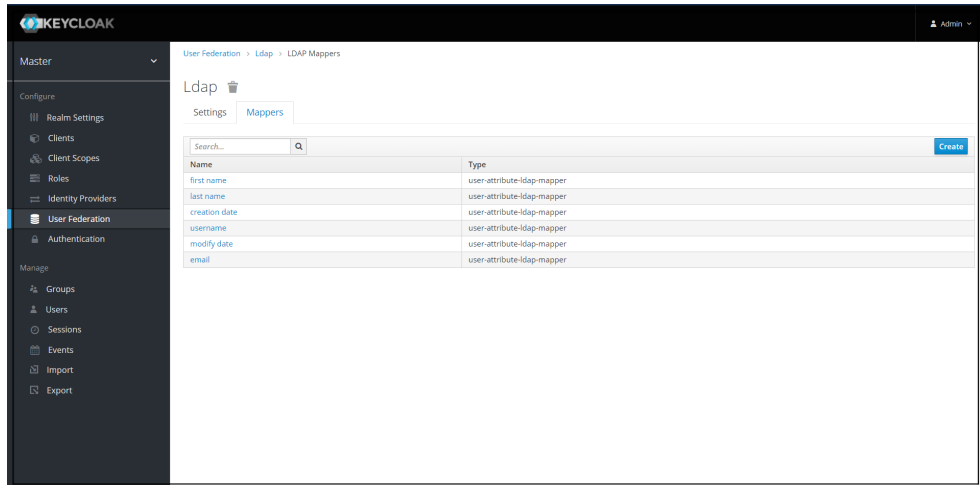
When an LDAP Federation provider is created, CNC Console-IAM provides a set of built-in mappers for this provider. User can change this set and create a new mapper or update/delete existing ones.

### Group Mapper

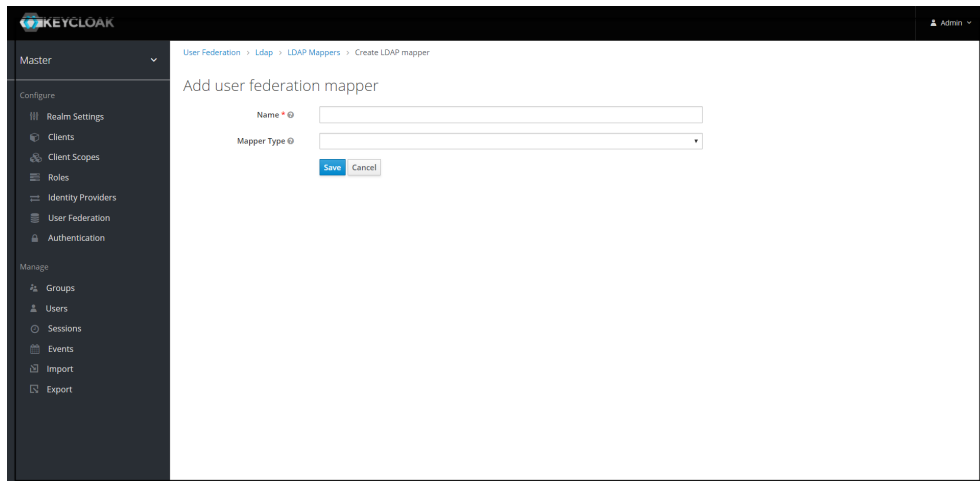
The Group Mapper allows you to configure group mappings from LDAP into cncc-iam group mappings. Group mapper can be used to map LDAP groups from a particular branch of an LDAP tree into groups in cncc-iam. It also propagates user-group mappings from LDAP into user-group mappings in cncc-iam.

**To add Group-Mapper and assign roles:**

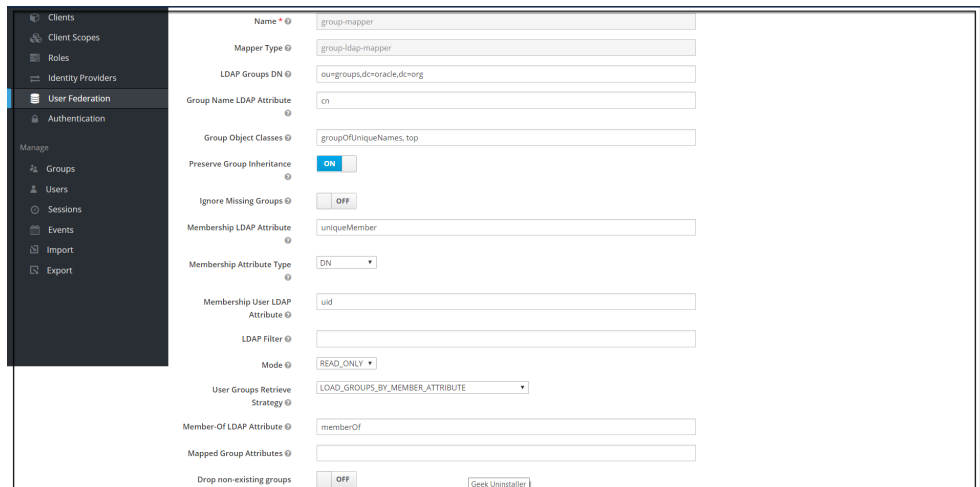
1. Click **Configure** and select **User Federation**. Click **Idap** (Console Display Name) and select the **Mappers** tab, and click **Create**.



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



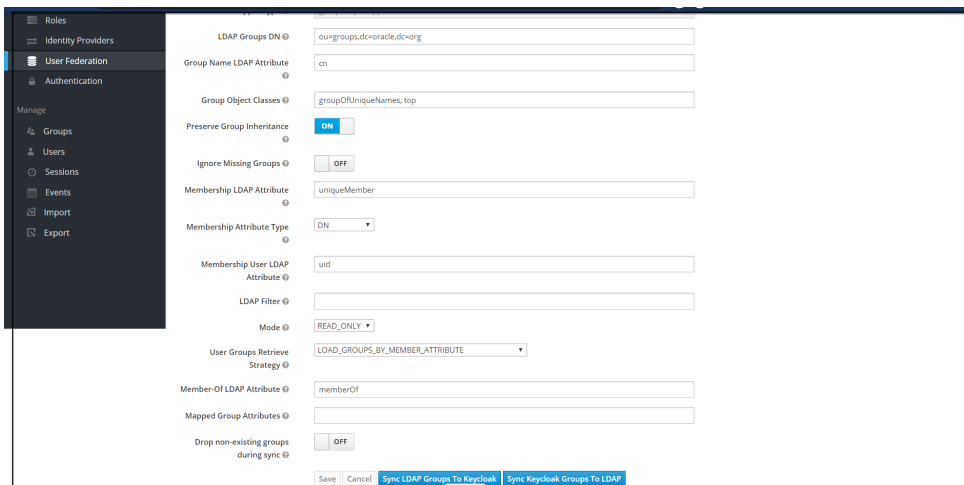
The following screen appears.



 **Note:**

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

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

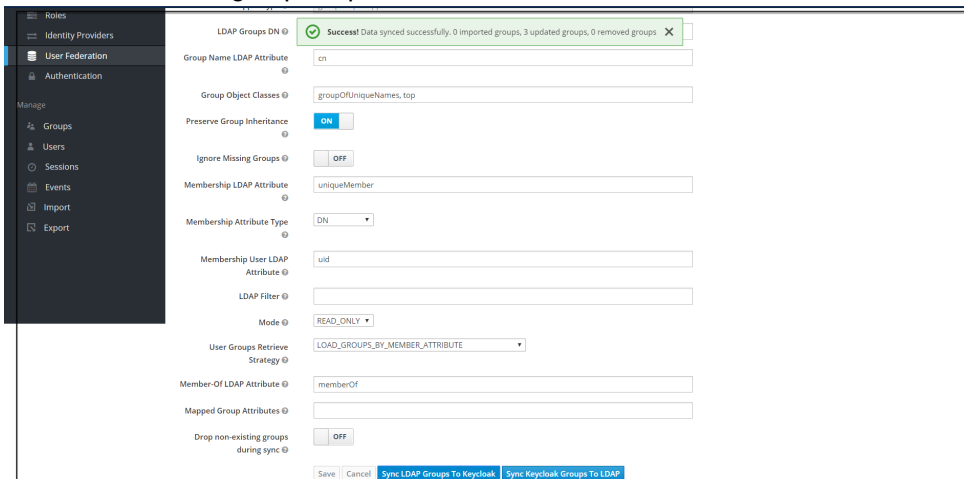


The screenshot shows the LDAP configuration interface. The left sidebar has a 'Manage' section with 'Groups' selected. The main panel contains the following fields and controls:

- LDAP Groups DN: ou=groups,dc=oracle,dc=org
- Group Name LDAP Attribute: cn
- Group Object Classes: groupOfUniqueNames, top
- Preserve Group Inheritance: ON
- Ignore Missing Groups: OFF
- Membership LDAP Attribute: uniqueMember
- Membership Attribute Type: DN
- Membership User LDAP Attribute: uid
- LDAP Filter: (empty)
- Mode: READ\_ONLY
- 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

At the bottom, there are buttons for 'Save', 'Cancel', 'Sync LDAP Groups To Keycloak', and 'Sync Keycloak Groups To LDAP'.

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

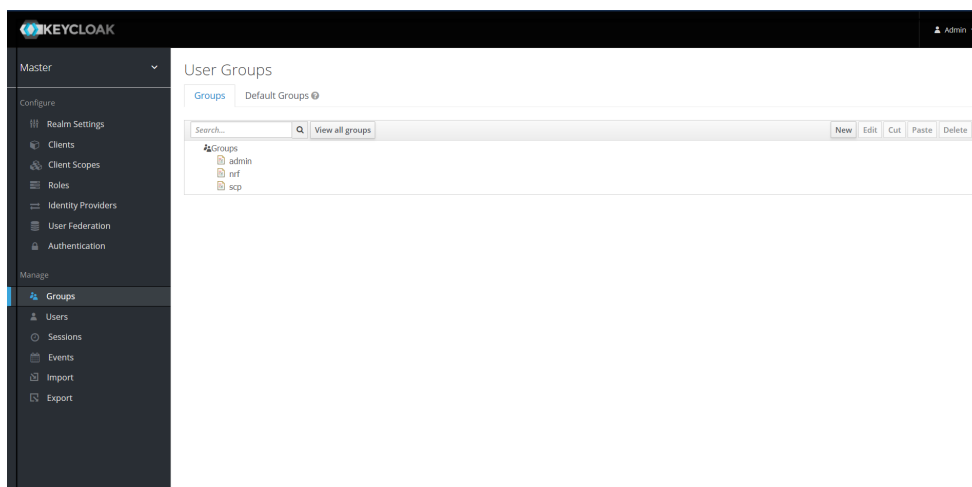


The screenshot shows the same LDAP configuration interface as above, but with a green success message at the top: "Success! Data synced successfully. 0 imported groups, 3 updated groups, 0 removed groups". The rest of the configuration fields and buttons remain the same.

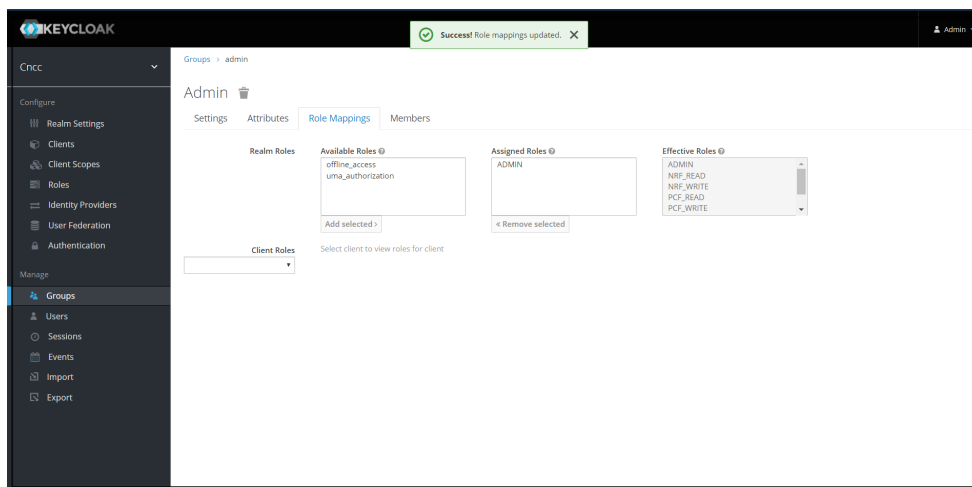
 **Note:**

If this step fails then you might need to check to the trouble shooting section and look at cnc-iam logs in debug mode.

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



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



- Select **Role Mapping** tab to see a list of roles that are pre-defined in cnc-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.

 **Note:**

For more information about the user roles, refer [APPENDIX](#).

# 5

## Accessing NF Resources through Curl or Postman

This section describes how CNC Console access NF resources through curl or postman:

CNC Console IAM provides a REST API for generating and refreshing access tokens. The API is used to get access token.

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

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

Example:

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

2. The body of the request must be *x-www-form-urlencoded* as given:

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

Example:

```
'client_id': 'cncc',  
'username': 'admin',  
'password': 'admin',  
'grant_type': 'password'
```

3. The Curl Command must be given. The Curl Command is as follows:

```
curl --location --request POST 'http://10.75.182.79:8080/cncc/auth/realms/cncc/protocol/openid-connect/token' \  
--header 'Content-Type: application/x-www-form-urlencoded' \  
--data-urlencode 'grant_type=password' \  
--data-urlencode 'username=shreb' \  
--data-urlencode 'password=Shreb123!' \  
--data-urlencode 'client_id=cncc'
```

4. As the response user gets an **access\_token** and a **refresh\_token**. The response is as follows:

```
{  
  "access_token":  
  "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXLTUwIiwiaWF0IjoiYjE6ICJHS1N4WVhoWlExRVh
```





# 6

## CNC Console Metrics

This section describes about CNCC Metrics.

### Common Metrics

Tags that are common across the metrics are described in the below table.

**Table 6-1 Common Metrics**

Dimension	Description	Possible Values
Method	Http method	GET, PUT, POST, DELETE, PATCH
NFType	Name of the NF Type.	UNKNOWN ( <b>Note:</b> Will be updated when Ingress is 5G aware)
NFServiceType	Name of the Service with in the NF.	UNKNOWN ( <b>Note:</b> Will be updated when Ingress is 5G aware)
Host	(Ip or fqdn) : port of ingress gateway	NA
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	N/A

### CNCC Core Ingress Gateway Metrics

**Table 6-2 CNCC Core Ingress Gateway Metrics**

S. No.	Metric Name	Metric Details	Metric Filter	Dimensions
1	Total Requests	Total number of requests received by Ingress Gateway for CNCC Core	oc_ingressgateway_ht tp_requests_total	<ul style="list-style-type: none"> <li>• NFType</li> <li>• NFServiceType</li> <li>• Host</li> <li>• HttpVersion</li> <li>• Scheme</li> <li>• Route_path</li> </ul>
2	Total Responses	Total number of responses sent by Ingress Gateway for CNCC Core requests	oc_ingressgateway_ht tp_responses_total	<ul style="list-style-type: none"> <li>• Status</li> <li>• Method</li> <li>• Route_path</li> <li>• NFType</li> <li>• NFServiceType</li> <li>• Host</li> <li>• HttpVersion</li> <li>• Scheme</li> </ul>

Table 6-2 (Cont.) CNCC Core Ingress Gateway Metrics

S. No.	Metric Name	Metric Details	Metric Filter	Dimensions
3	Success (2xx) Responses	Total number of success responses (2xx) sent by Gateway for CNCC Core requests	oc_ingressgateway_ht tp_responses_total{St atus=~"2.*"}	<ul style="list-style-type: none"> <li>• Status</li> <li>• Method</li> <li>• Route_path</li> <li>• NFType</li> <li>• NFServiceType</li> <li>• Host</li> <li>• HttpVersion</li> <li>• Scheme</li> </ul>
4	Error (5xx) Responses	Total number of error responses (5xx) sent by Gateway for CNCC Core requests	oc_ingressgateway_ht tp_responses_total{St atus=~"5.*"}	<ul style="list-style-type: none"> <li>• Status</li> <li>• Method</li> <li>• Route_path</li> <li>• NFType</li> <li>• NFServiceType</li> <li>• Host</li> <li>• HttpVersion</li> <li>• Scheme</li> </ul>
5	Error (4xx) Responses	Total number of error responses (4xx) sent by Gateway for CNCC Core requests	oc_ingressgateway_ht tp_responses_total{St atus=~"4.*"}	<ul style="list-style-type: none"> <li>• Status</li> <li>• Method</li> <li>• Route_path</li> <li>• NFType</li> <li>• NFServiceType</li> <li>• Host</li> <li>• HttpVersion</li> <li>• Scheme</li> </ul>
6	Request Body	Total number of requests with request body	oc_ingressgateway_re quest_content_metr ics_total	
7	Request Processing Time Ingress	Time taken for processing the request by Ingress gateway.	oc_ingressgateway_re quest_processing_la tency	quantile
8	Request Processing Time Total	Total time taken for processing the request	oc_ingressgateway_re quest_latency	quantile

### CNCC IAM Ingress Gateway Metrics

Table 6-3 CNCC IAM Ingress Gateway Metrics

S. No.	Metric Name	Metric Details	Metric Filter	Dimensions
1	Total Requests	Total number of requests received by Ingress Gateway for CNCC IAM	oc_ingressgateway_ht tp_requests_total	<ul style="list-style-type: none"> <li>• NFType</li> <li>• NFServiceType</li> <li>• Host</li> <li>• HttpVersion</li> <li>• Scheme</li> <li>• Route_path</li> </ul>

Table 6-3 (Cont.) CNCC IAM Ingress Gateway Metrics

S. No.	Metric Name	Metric Details	Metric Filter	Dimensions
2	Total Responses	Total number of responses sent by Ingress Gateway for CNCC IAM requests	oc_ingressgateway_ht tp_responses_total	<ul style="list-style-type: none"> <li>• Status</li> <li>• Method</li> <li>• Route_path</li> <li>• NFType</li> <li>• NFServiceType</li> <li>• Host</li> <li>• HttpVersion</li> <li>• Scheme</li> </ul>
3	Success (2xx) Responses	Total number of success responses (2xx) sent by Gateway for CNCC IAM requests	oc_ingressgateway_ht tp_responses_total{St atus=~"2.*"}	<ul style="list-style-type: none"> <li>• Status</li> <li>• Method</li> <li>• Route_path</li> <li>• NFType</li> <li>• NFServiceType</li> <li>• Host</li> <li>• HttpVersion</li> <li>• Scheme</li> </ul>
4	Error (5xx) Responses	Total number of error responses (5xx) sent by Gateway for CNCC IAM requests	oc_ingressgateway_ht tp_responses_total{St atus=~"5.*"}	<ul style="list-style-type: none"> <li>• Status</li> <li>• Method</li> <li>• Route_path</li> <li>• NFType</li> <li>• NFServiceType</li> <li>• Host</li> <li>• HttpVersion</li> <li>• Scheme</li> </ul>
5	Error (4xx) Responses	Total number of error responses (4xx) sent by Gateway for CNCC IAM requests	oc_ingressgateway_ht tp_responses_total{St atus=~"4.*"}	<ul style="list-style-type: none"> <li>• Status</li> <li>• Method</li> <li>• Route_path</li> <li>• NFType</li> <li>• NFServiceType</li> <li>• Host</li> <li>• HttpVersion</li> <li>• Scheme</li> </ul>
6	Request Body	Total number of requests with request body	oc_ingressgateway_re quest_content_metr ics_total	
7	Request Processing Time Ingress	Time taken for processing the request by Ingress gateway.	oc_ingressgateway_re quest_processing_lat ency	quantile
8	Request Processing Time Total	Total time taken for processing the request	oc_ingressgateway_re quest_latency	quantile

# 7

## CNCC Logs

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

[Log Formats and Details](#)

[Types of Logs](#)

[Configuring the Logs](#)

[Examples of Logs](#)

[Accessing the Logs](#)

### Log Formats

This section describes 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": <threadId>,
  "level": <log_level>,
  "loggerName": <name_of_the_logging_class>,
  "message": <message>,
  "instant": <timestamp_in_miliseconds>,
  "messageTimestamp": <timestamp_in_readable_format>
  "threadId": <threadId>,
  "threadPriority": <threadPriority>,
  "pod": <name_of_the_pod>,
  "processId": <processId>,
  "contextMap": <context_map>
}
```

**Table 7-1 Log Details**

Name	Description	Example
<b>thread</b>	Name of the thread	"thread": "reactor-http-epoll-1"
<b>level</b>	Level of the log. It can be: Log level (INFO, WARN, DEBUG, TRACE)	"level": "INFO"
<b>loggerName</b>	Name of the class that generated the log	"loggerName": "ocpm.cne.gateway.cncc.GatewayApplication"



Table 7-1 (Cont.) Log Details

Name	Description	Example
<b>messageTimestamp</b>	Time represented in human readable format and in UTC.	"messageTimestamp": 2020-07-04 12:00:40.702
<b>message</b>	Information about the event	"message": "Started Application....." By default, all messages are in simple string except <b>Audit Log, Security Log</b> which are represented in CNCC Message Format.
<b>instant</b>	The Date and Time the event occurred in epoch second and nano seconds	"instant": { "epochSecond": 1590045388, "nanoOfSecond": 339789000}
<b>processId</b>	Linux process Identifier (for a multi- process host)	Linux process Identifier (for a multi-process host).
<b>threadId</b>	Id of the thread	"threadId": "43"
<b>threadPriority</b>	Priority assigned to the thread	"threadPriority": 5
<b>pod</b>	Name of the pods where the log is generated	"cncc-core-ingress- gateway-77df795fb5-wv2sb"
<b>contextMap</b>	It hold information that are added to threadContext.	"contextMap": { "hostname": "cncc-core- ingress-gateway-77df795fb5- wv2sb", "ingressTxId": "ingress- tx-1460885598"}

## CNCC Message Format

Table 7-2 CNCC Message Format

Name	Description	Example	Possible Values
<b>logType</b>	Indicates whether it is <b>Security Log</b> or <b>Audit Log</b>	logType=AUDIT	AUDIT SECURITY
<b>type</b>	Indicates nature/action of the log	type=REQUEST	For Security Log, REQUEST RESPONSE For Audit Log, LOGIN ACCESS_RESOURCE ACCESS_RESOURCE_E RROR LOGOUT

Table 7-2 (Cont.) CNCC Message Format

Name	Description	Example	Possible Values
<b>resourceType</b>	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 NF's
<b>userId</b>	Id of the user. Basically to know who triggered request/action	userId=3314f54f-08bf-489d-b395-27bf56da1262	
<b>username</b>	Name of the user	username=shreb	
<b>status</b>	Http status of the response.	status=200 OK	
<b>operationType</b>	HTTP method of the request	operationType=GET	
<b>scheme</b>	Indicates the scheme of the request	scheme=http	
<b>remoteAddress</b>	Remote Address associated with request. i.e remote address where this request is connected to, if available.	remoteAddress=/192.168.219.64:53587	
<b>localAddress</b>	Local Address associated with request. i.e local address the request was accepted on, if available.	localAddress=cncc-core-ingress-gateway.cncc.svc.cluster.local/<unresolved>:30075	
<b>resourcePath</b>	Request uri	resourcePath=/soothsayer/v1/canaryrelease/	
<b>queryParams</b>	Query parameters associated with request	queryParams={form_id=9, page=1, view_id=78}	

Table 7-2 (Cont.) CNCC Message Format

Name	Description	Example	Possible Values
<b>headers</b>	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}	
<b>payload</b>	Payload/Data associated with request or response	payload=[{"serviceName":"n5g-eir-eic","canaryReleaseFlag":true,"apiFullVersion":"2.0.0","canaryTraffic":5}]	
<b>authentication Type</b>	This indicates whether user is requesting resource logged in using CNCC or directly accessing through postman/curl.	authenticationType=OAUTH	OAUTH -> User is logged in through CNCC application and accessing resource JWT -> User is accessing resource directly through postman/curl

## Types of Logs

The CNCC logs can be categorized into following types:

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

### Regular logs

These logs contains all kinds of error messages, warnings or other events written within the application which provide logical, highlevel 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 contains user related information and his activity within the system.

Following events are logged in CNCC Core:

- Login - A user has logged in.
- Access Resource- A user is accessing particular NF resource.
- Access Resource Error - A user is denied from accessing 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

Following events are logged in CNCC IAM:

#### Login events

- Login - A admin user has logged in.
- Register - A admin user has registered.
- Logout - A admin user has logged out.
- Code to Token - An application/client has exchanged a code for a token.
- Refresh Token - An application/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* and are represented in Keycloak format. These events are provided by keycloak and are documented under [Keycloak Auditing End Events](#).

### Security Logs

The security logs holds information of all the request and its corresponding response. Information such as header, payload, method, scheme, uri etc.

At INFO level it logs,

- only those request, with header *Content-Type/Accept* is set to *application-json/www-form-urlencoded*
- corresponding response, with header *Content-Type* is set to *application-json/prolem+json/www-form-urlencoded*

At DEBUG level it logs,

- all request.
- all response.

### Request/Response Payload

At all the log levels:

- payload is logged only for request/response with header *Content-Type/Accept* is set to *application-json/prolem+json/www-form-urlencoded*
- all *html, css, javascript, icon, woff* payload are masked.

### Header Information

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



#### Note:

The user can find the Security logs :

- For CNCC Core, these logs are logged as part of *cncc-core-ingress-gateway* and are represented in CNCC message Format.
- For CNCC IAM, these logs are logged as part of *cncc-iam-ingress-gateway* and are represented in CNCC message Format.

## Configuring Security Logs

This section details about configuring security logs.

### Setting at Log Level

By default **Security Log** will be set to "INFO" level for both *CCNC Core* and *CNCC IAM*. But user can change it log level by setting *log.level.cncc.security* to required level in core and iam *values.yaml*

### values.yaml

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

```
cncc:
  security: INFO
```

### Disabling Security Log

By default **Security Log** will be enabled for both *CCNC Core* and *CNCC IAM*. But user can disable this by setting *securityLogEnabled* flag to **false** in *core/iam values.yaml*

### values.yaml

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

## Examples of Logs

This section lists the examples of audit and security logs.

### Examples of Audit Logs

### Examples of Security Logs

### Examples of Audit Logs

#### CNCC Core

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

- User successfully logging into CNCC Core

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

- User accessing SCP resource having SCP\_READ role

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

- User updating(PATCH) SCP resource having SCP\_WRITE role

```
logType=AUDIT, type=ACCESS_RESOURCE, resourceType=SCP,
userId=186f6f2a-ba6a-4812-8a18-b906a5f9e3f6, username=shreb,
operationType=PATCH, remoteAddress=/192.168.219.64:53587,
localAddress=cncc-core-ingress-gateway.cncc.svc.cluster.local/
<unresolved>:30075,
resourcePath=/soothsayer/v1/canaryrelease/n5g-eir-eic,
authenticationType=OAUTH
```

- User accessing NRF resource without having NRF\_READ role

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

- User successful logout

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

### CNCC IAM:

- Login Error when password entered was wrong

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

- Login with correct credential

```
04:57:24,581 INFO [org.keycloak.events] (default task-22)
                                type=LOGIN, realmId=master,
clientId=security-admin-console,
                                userId=d7cde46f-15e1-4ff8-a2cb-
c5825e481438, ipAddress=192.168.219.64,
                                auth_method=openid-connect,
auth_type=code, redirect_uri=http://10.75.225.28:31373/cncc/auth/admin/master/
console/, consent=no_consent_required,
                                code_id=5aca4960-eecf-406b-
a7eb-92e249c2beeb, username=admin,
                                authSessionParentId=5aca4960-
eecf-406b-a7eb-92e249c2beeb,
                                authSessionTabId=8sruELAlWws
```

- User created

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

- Deleted user

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

- Admin Role removed for a user

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



- Admin Role added for a user

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

- Realm setting update

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

- Logout all session on keycloak

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

## Examples of Security Logs

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

### CNCC Core

- SCP request

```
logType=SECURITY, type=REQUEST, resourceType=SCP,
                    userId=3314f54f-08bf-489d-
b395-27bf56da1262, username=shreb,
                    operationType=GET, scheme=http,
                    remoteAddress=/
192.168.219.64:53587,
                    localAddress=cncc-core-ingress-
gateway.cncc.svc.cluster.local/<unresolved>:30075,
                    resourcePath=/soothsayer/v1/
canaryrelease/, queryParams={},
                    headers={Accept=*/, X-
Requested-With=XMLHttpRequest,
                    User-Agent=Mozilla/5.0 (Windows
```

```

NT 10.0; WOW64; rv:68.0)
Gecko/20100101 Firefox/68.0,
Connection=keep-alive,
Host=cncc-core-ingress-
gateway.cncc.svc.cluster.local:30075,
Accept-Language=en-US,en;q=0.5,
Accept-Encoding=gzip, deflate,
DNT=1, Content-Type=application/
json; charset=utf-8},
payload={},
authenticationType=OAUTH

```

- SCP response

```

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

```

## CNCC IAM

- Request

```

logType=SECURITY, type=REQUEST, operationType=GET, scheme=http,
remoteAddress=
192.168.219.64:53587,
localAddress=cncc-iam-ingress-
gateway.cncc.svc.cluster.local/<unresolved>:30085,
resourcePath=/cncc/auth/admin/

```

```

master/console/config,
headers={Accept=application/json,
NT 10.0; WOW64; rv:68.0)
68.0, Referer=http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/cncc/auth/
admin/master/console/, Connection=keep-alive,
gateway.cncc.svc.cluster.local:30085,
Accept-Encoding=gzip, deflate,
Accept-Language=en-US,en;q=0.5,
DNT=1}, payload={},
authenticationType=NONE
queryParams={},
User-Agent=Mozilla/5.0 (Windows
Gecko/20100101 Firefox/

```

- **Response**

```

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

```

## Accessing logs

This section gives information about how to access the logs.

The CNCC application logs can be accessed in following ways:

1. Viewing logs of a cncc application pod running. This can be achieved by executing the command:

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

Example:

CNCC Core:

```
$ kubectl logs -f -n cncc cncc-core-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)
```

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

Example : EFK can be used here with CNCC to view the logs as given below.



# A

## CNC Console Roles

### Overview

Access management for resources is a critical function for any organization.

Role Based Access Control (RBAC) helps in:

- Access Management
- Resource Management
- Managing user access to resources
- Managing user access to areas

### Role Based Access Control

RBAC restricts network access based on a person's role within an organization and has become one of the main methods for advanced access control. The roles in RBAC refer to the levels of access that employees have in the network.

### Role

A role is a collection of permissions that you can apply to users. Using roles makes it easier to add, remove, and adjust permissions than assigning permissions to users individually.

As the user base increases in scale and complexity, roles become particularly useful.

### Composite Role

A Composite Role is a role that has one or more additional roles associated with it. When a composite role is mapped to the user, the user also gains the roles associated with that composite.

## Types of Roles in CNC Console

In CNCC, Role Based Access Control (RBAC) is controlled by third-party Identity Access Management (IAM) provider called **Keycloak**. Roles related to CNCC applications are defined in IAM.

Roles are predefined for CNCC application.

Roles are of 2 categories.

1. **ADMIN**

2. **NF**

**ADMIN:**

Role: **ADMIN**

User having this role has access to all resources (NF resources) within CNCC application.

Allowed Operations: CREATE, READ, UPDATE, DELETE

Composite Roles: All NF Level roles.

Example: If a user has ADMIN role, then the user can read, create, update, or delete any MOs configurations of any NFs that is supported by CNCC application.

**NF:**

NF level roles are divided further into:

1. **<NF>\_READ**
2. **<NF>\_WRITE**

 **Note:**

<NF> is placeholder. Say for example, if CNCC supports POLICY and SCP NFs then, POLICY\_READ, POLICY\_WRITE, SCP\_READ and SCP\_WRITE roles would be defined for CNCC application in IAM.

**Role: <NF>\_READ**

User having this role can only read configurations from all Managed Objects (MOs) within particular NF.

Allowed Operations: READ

NFs: One particular NF.

Composite Roles: No roles.

Example: If user has POLICY\_READ then the user:

- Can only read configurations of any MOs configurations within the NF.
- Cannot write/update/delete any record.

**Role: <NF>\_WRITE**

User having this role has access one particular NF and can perform CRUD operations.

Allowed Operations: CREATE, READ, UPDATE, DELETE

NFs: One particular NF.

Composite Roles: <NF>\_READ role.

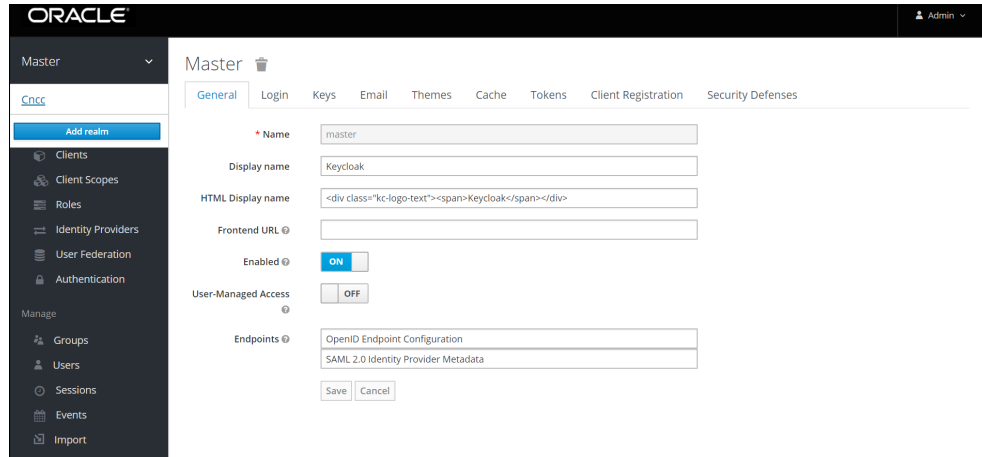
Example: If user has POLICY\_WRITE then the user can read/write/update/delete any MOs configurations within the NF.

## How to Set or Update User Password in CNCC IAM

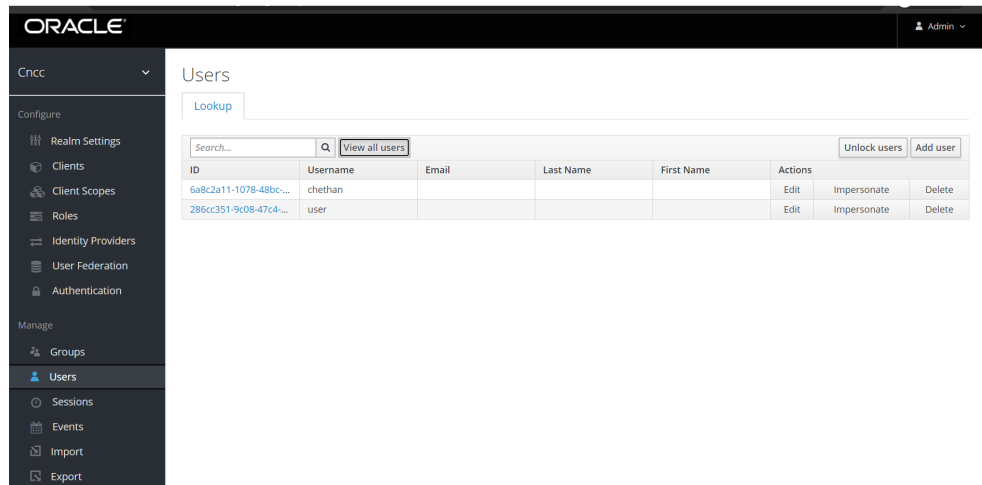
This section describes about Setting or updating the user password in CNCC IAM.

The steps to set or update the user password are as below:

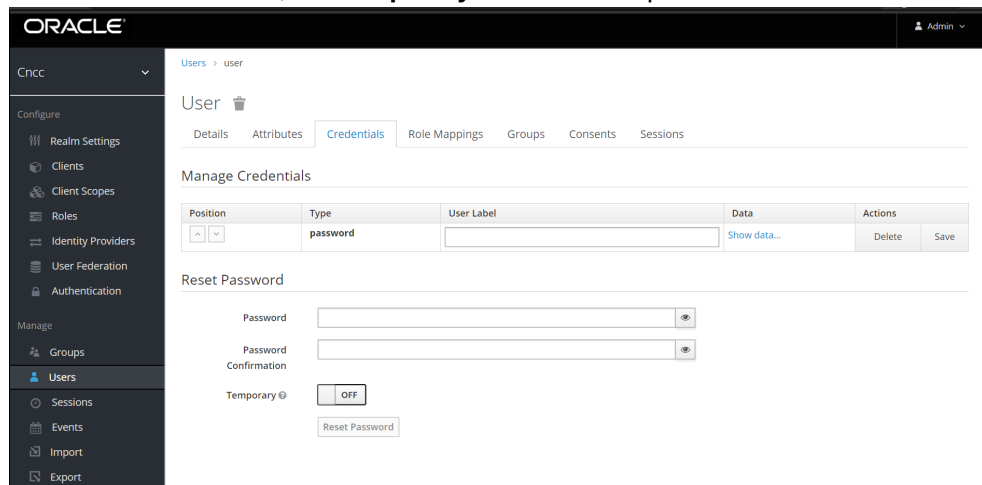
1. Select the **Cncc** Realm.



2. Select **Users** on the left pane and click **view all users** at the right pane. Click **Edit** under **Actions** to update the credentials.



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

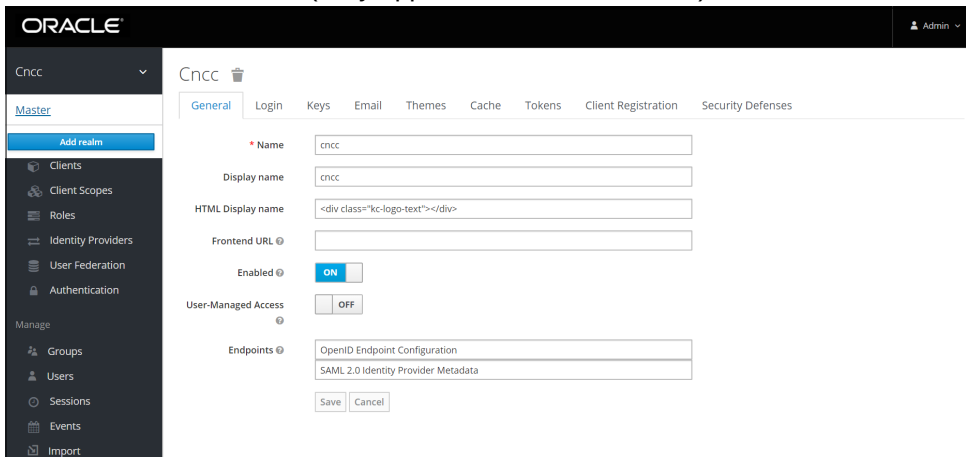


## How to Set or Update Admin Password in CNCC IAM

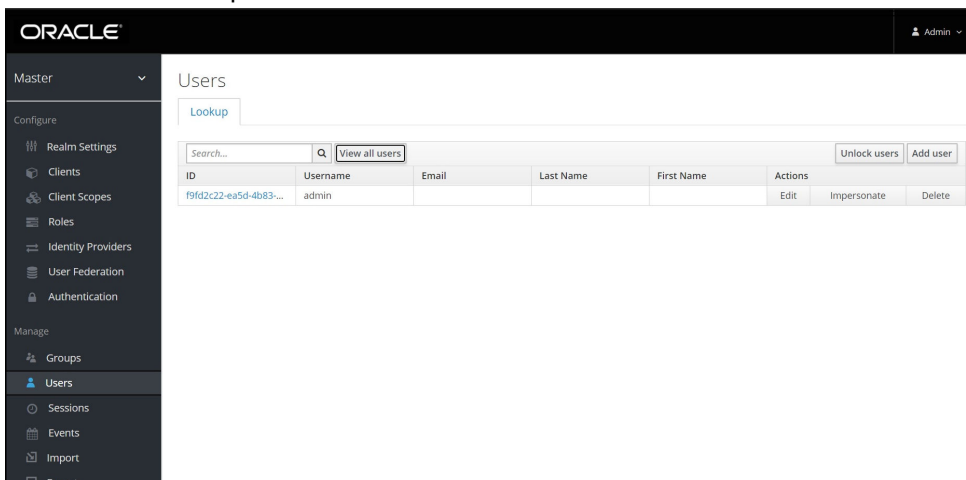
This section describes about Setting or updating the admin password in CNCC IAM.

The steps to set or update the admin password are as below:

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



2. Select **Users** on the left pane and click **view all users** at the right pane. Click **Edit** under **Actions** to update the admin credentials.



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

