Oracle® Communications

Cloud Native Core Console User's Guide

Release 1.1.0
F32118-01
May 2020

ORACLE

Oracle Communications Cloud Native Core Console User's Guide, Release 1.1.0

F32118-01

Contents

List of Tables

1

What's New in This Guide

- The Network Function **UDR 1.6** is integrated to the GUI
- The Network Function **CNPCRF 1.6** is integrated to the GUI
- The Network Functions **NRF** and **PCF** is upgraded to version 1.6.1
- The Network Function **SCP** is upgraded to version 1.6

2 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)
- Policy Control Function (PCF)
- Service Communication Proxy (SCP)
- Cloud Native Policy and Charging Rules Function (CNPCRF)
- Unified Data Repository (UDR)

This document gives a brief idea about configuring NRF, PCF, SCP, CNPCRF 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:

- View Roles in CNC Console IAM
- Create Users (Applicable only if not Integrating with LDAP)
- View the Users
- Assign Roles to User (Applicable only if not Integrating with LDAP)
- LDAP Server integration in CNC Console IAM
- Access NF Resources through curl or postman

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 Policy Control Function User's Guide
- Cloud Native Policy and Charging Rules Function (CNPCRF) 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
- Cloud Native Policy Control Function Installation Guide
- Unified Data Repository (UDR) Cloud Native Installation and Upgrade Guide

- Cloud Native Policy and Charging Rules Function (CNPCRF) Installation Guide

Acronyms

Table 2-1 Acronyms

Terms	Definition
CNCC	Cloud Native Core Console
NRF	Network Repository Function
OSDC	Oracle Software Delivery Cloud
SCP	Service Communication Proxy
PCF	Policy Control Function
IAM	Identity Access Management
UDR	Unified Data Repository
CNPCRF	Cloud Native Policy and Charging Rules Function
LDAP	Lightweight Directory Access Protocol
HTTPS	Hypertext Transfer Protocol Secure

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.

3
CNC Console

This section provides information about CNC Console.

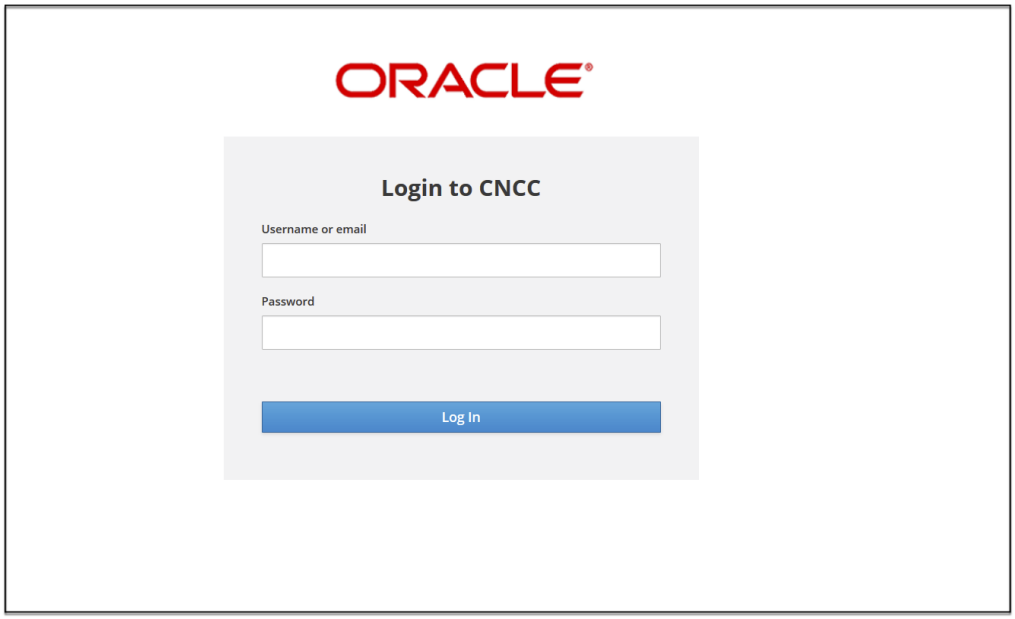
The CNC Console has two modules:

- **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 cncc-iam-ingress-ip* and *<port number> is cncc-iam-ingress-port*

3. Enter the valid credentials.

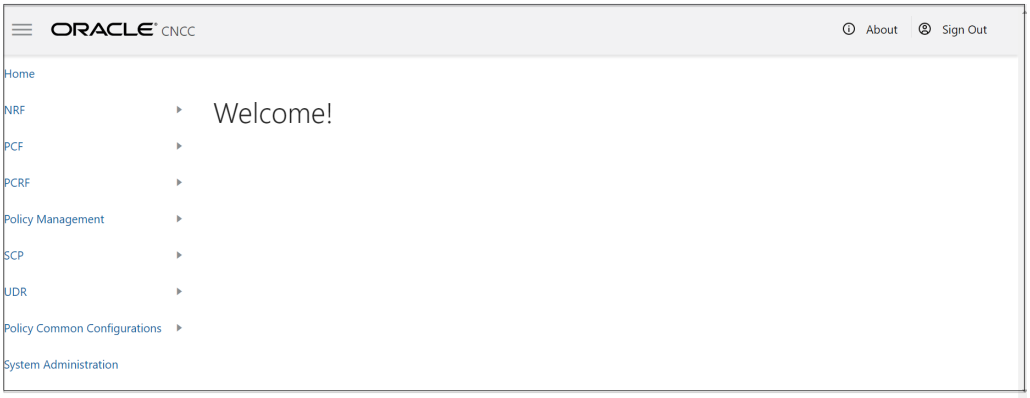
4. Click **Log In**. The Welcome Screen of CNC Console interface appears.

> **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 Collapse button at the left side allows the user to collapse the left pane.

Types of User Interface Screens for NF Configurations

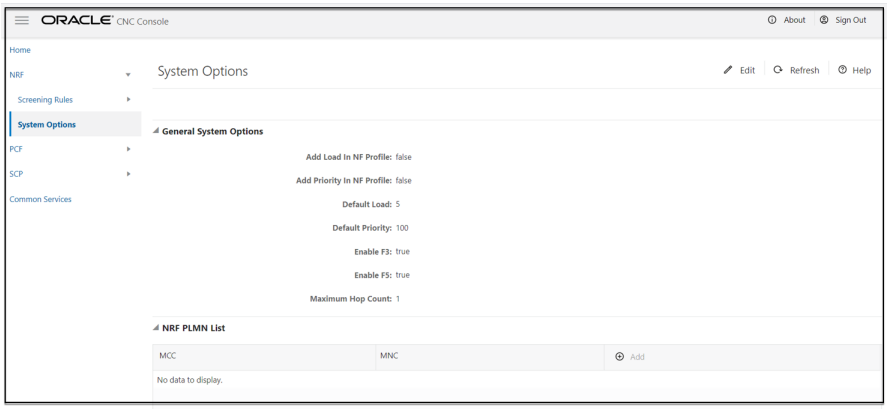
CNC Console Screens

CNC Console has two types of screens:

Service Screen

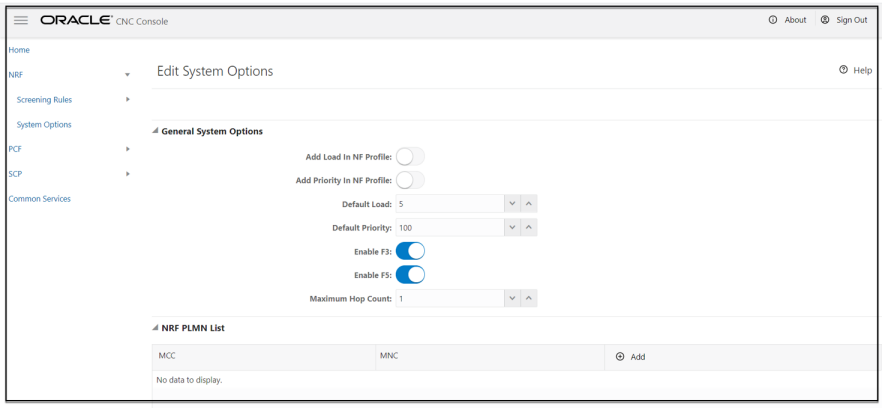
Service Screens has single independent objects. These kinds of screen are used to display and configure single object.
Examples for the Service Screen:

Service Screen



Service Screen - Edit

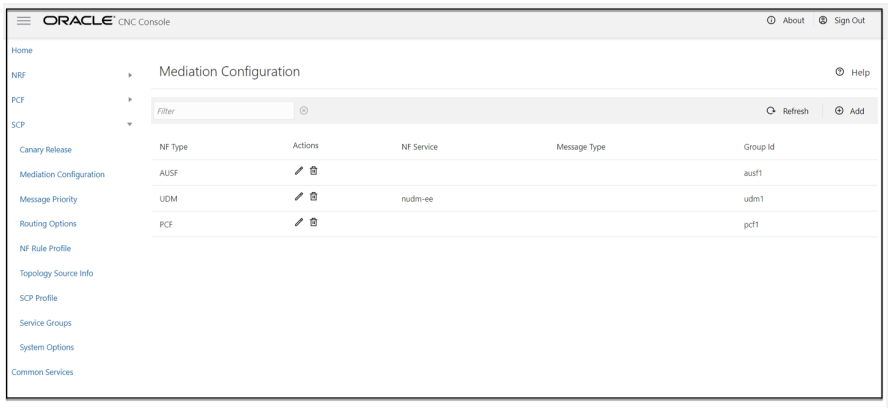
The Service screen with **Edit** enabled. The user can update the parameters.



Configurations Screen

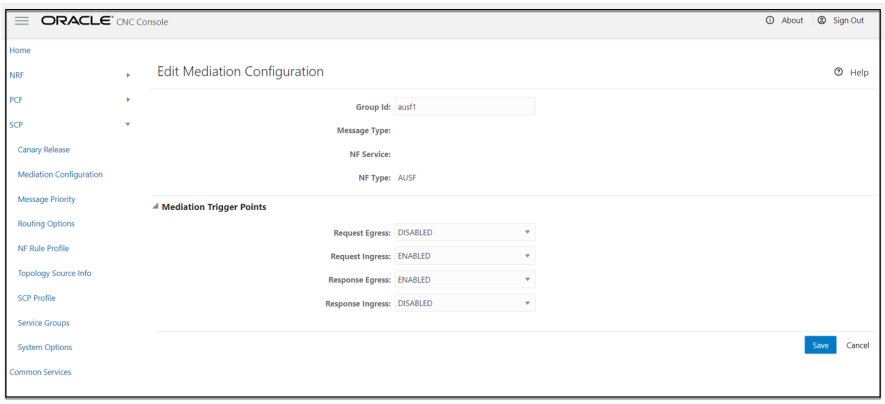
The Configurations screen is used to display and configure multiple related objects.

Examples for the Configuration Screen:



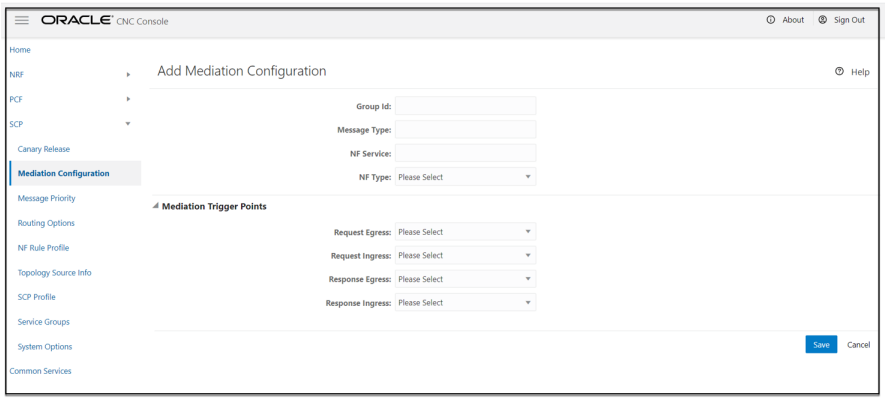
Configurations Screen- Edit

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



Configurations Screen - Add

The Configurations screen with **Add** enabled.



4

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

1. Click **NRF** on the left navigation pane. The **Screening Rules** and **System options** appear underneath.
2. 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.**
3. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
4. Click **Edit** modify the parameters.
5. On selecting **System Options,** the parameters of **System Options** appear on the right pane.
6. Click **Edit** to modify the parameters.
7. Click **Save**.

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

Policy Control Function (PCF)

Overview

The Oracle Communications Policy Management solution is enhanced to add Policy Control Function that extends the functionality of PCRF as part of 5G core network. The Policy Control Function is a functional element for policy control decision and flows based charging control functionalities. The PCF provides the following functions:

- Policy rules for application and service data flow detection, gating, QoS, and flow based charging to the SMF.
- Access and Mobility Management related policies to the AMF.

Configuring PCF Parameters

1. Click **PCF** on the left navigation pane. The **Global Configurations, Service Configurations, Policy Configurations** and **Session Viewer** appear underneath.
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 different PCF services appear underneath the **Service Configurations**. Different services are: the functionalities of **Service Configurations** appear underneath. The functionalities are **Session Management Service, Access and Mobility Service, User Service, Policy Authorization Service** and **UE Policy Service**.
5. On selecting a service, the configurable parameters of the functionality appear on the right pane.
6. Click **Edit** to modify the parameters.
7. On selecting **Policy Configurations**, the policy types of **Policy Configurations** appear underneath. The policy types are **Common, SM Policy, AM Policy** and **UE Policy**.
8. 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.
9. Click **Edit** to modify the parameters.
10. On selecting **Session Viewer**, the parameters of **Session Viewer** appear on the right pane.
11. Click **Edit** to modify the parameters.
12. Click **Save**.

> **Note:**
>
> For details refer to **Cloud Native Policy Control Function 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

1. Click **SCP** on the left navigation pane. The **Canary Release, Mediation Configuration, Message Priority, Routing Options, NF Rule Profile, Topology Source Info, SCP Profile, Service Groups** and **System Options** appear underneath.
2. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
3. Click **Edit** to modify the parameters.
4. 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

1. Click **UDR** on the left navigation pane. The **Provisioning** functionality appear underneath.
2. On selecting **Provisioning**, the functionalities of **Provisioning** appear underneath. The functionalities are **Profile Data, PCF, SLF** and **UDM**.
3. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
4. Click **Edit** to modify the parameters.
5. Click **Save**.

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

Cloud Native Policy and Charging Rules Function (CNPCRF)

Overview

The Oracle Communications Cloud Native Policy and Charging Rules Function (CNPCRF) solution incorporates new architecture with spring microservice framework as backend support technology stack and Kubernetes Cloud Native Environment as running environment. The PCRF core service is the main functionality among PCRF micro services with the following enhancements when compared to legacy PCRF:

- Remove the MIA module from MPE, and let the MPE talks to with configuration server to save/load related data.
- PCRF core service have integrated the MPE functionalities which are under legacy PCRF.
- When PCRF Core needs to talk with any data source, these traffic shall go with the Diameter connector rather than from the PCRF core itself.

Configuring CNPCRF Parameters

1. Click **CNPCRF** on the left navigation pane. The **Configurations, Session Viewer** and **Services** appear underneath.
2. On selecting **Configurations**, the manageable objects of **Configurations** appear underneath. The manageable objects are **Charging Server, Custom AVP, Custom Vendor, Logs, Media Profile, Network Element, PRA, Policy Counter Id, Retry Profile, Serving Gateway, Time Periods** and **Traffic Profile.**
3. On selecting a manageable object, the configurable parameters of that manageable object appear on the right pane.
4. Click **Edit** to modify the parameters.
5. On selecting **Session Viewer**, the parameters of **Session Viewer** appear on the right pane.
6. Click **Edit** to modify the parameters.
7. On selecting **Services**, the **Core Service** appears underneath.
8. Clicking on **Core Service** the parameters of **Core Service** appear on the right pane.
9. Click **Edit** to modify the parameters.
10. Click **Save.**

> **Note:**
>
> For details about configurable parameters, refer to **Cloud Native Policy and Charging Rules Function User's Guide.**

Policy Management and Policy Common Configurations

Policy Management and Policy Common Configurations are common for PCF and CNPCRF NFs.

Policy Management

Overview
User can create and manage a policy project for each of the CNPCRF and PCF services by using Policy Management.

Configuring Policy Management Parameters

1. Click **Policy Management** on the left navigation pane.The **Settings**, **Policy Projects, Policy Table, Policy Library, Test Policy Projects** and **Dropdown Blocks** appear underneath.
2. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
3. Click **Edit** to modify the parameters.
4. After updating the required parameters, click **Save.**

Policy Common Configurations

Overview
User can configure the managed objects which are common to PCF and CNPCRF by using Policy Common Configurations.

Configuring Policy Common Configurations

1. Click Policy Common Configurations on the left navigation pane. The **Diameter Configurations, Data Source Configurations, Subscriber Logging, Custom Attributes, Match List** and **Bulk Import** appear underneath.
2. On selecting a functionality, the configurable parameters of the functionality appear on the right pane.
3. Click **Edit** to modify the parameters.
4. After updating the required parameters, click **Save.**

> **Note:**
>
> For details refer to **Policy Control Function (PCF) Cloud Native User's Guide** and **Cloud Native Policy and Charging Rules Function (CNPCRF) User's Guide**.

Network Functions and Versions

Following are the supported NF versions for CNC Console 1.1.0:

NF	NF Version
SCP	1.6.0
NRF	1.6.1
PCF	1.6.1
CNPCRF	1.6.0
UDR	1.6.0

5
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 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

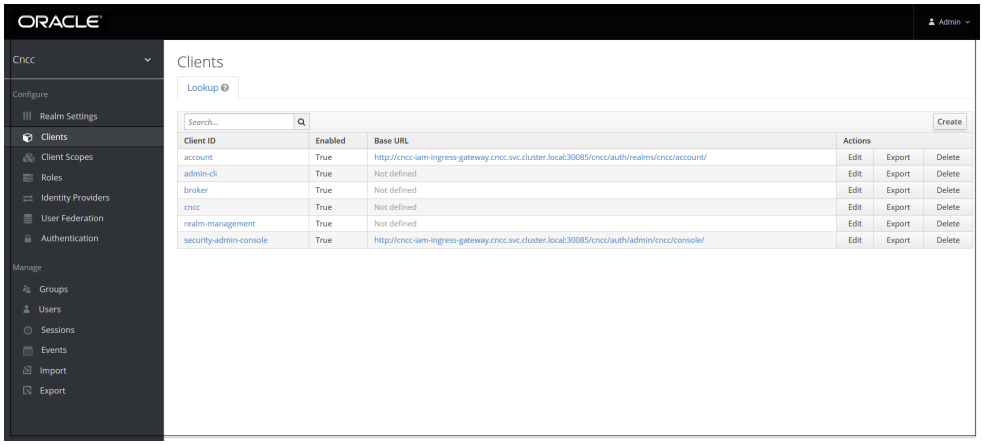
Once CNCC IAM is deployed administrator must do the setting of cncc redirection URL:

To set up the cncc redirection URL:

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

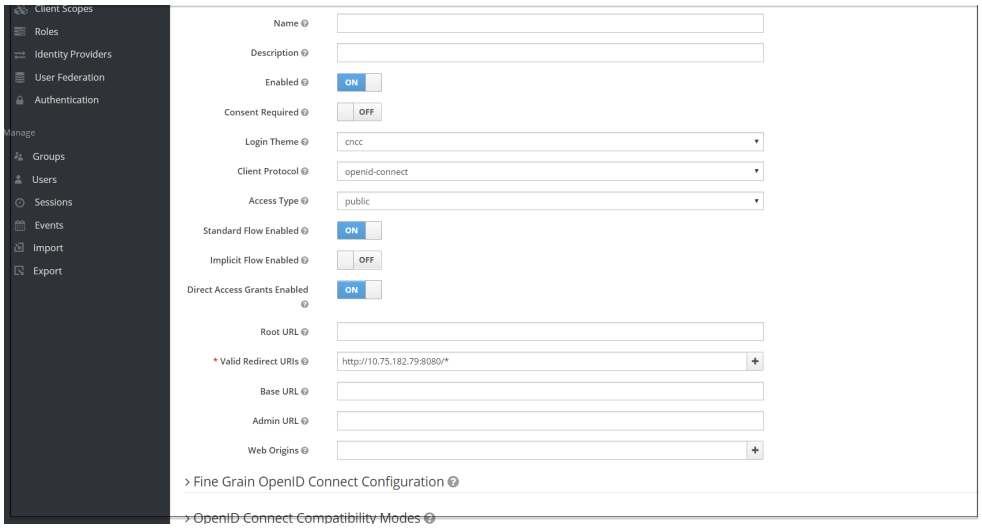
   ```
   <scheme>://<cncc-iam-ingress-extrenal-ip>:<cncc-iam-ingress-service-port>

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

2. Go to **Clients** and select **Cncc**.



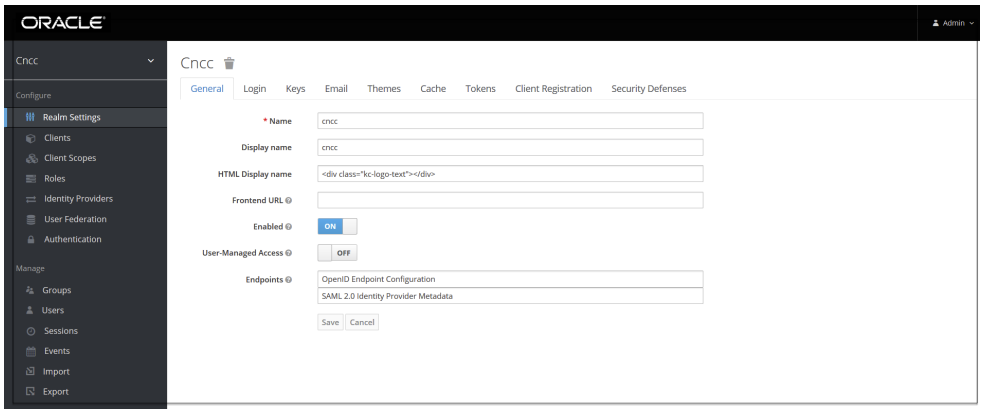
3. Enter CNCC Core Ingress URI in the **Valid Redirect URIs** field and click **Save.**

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

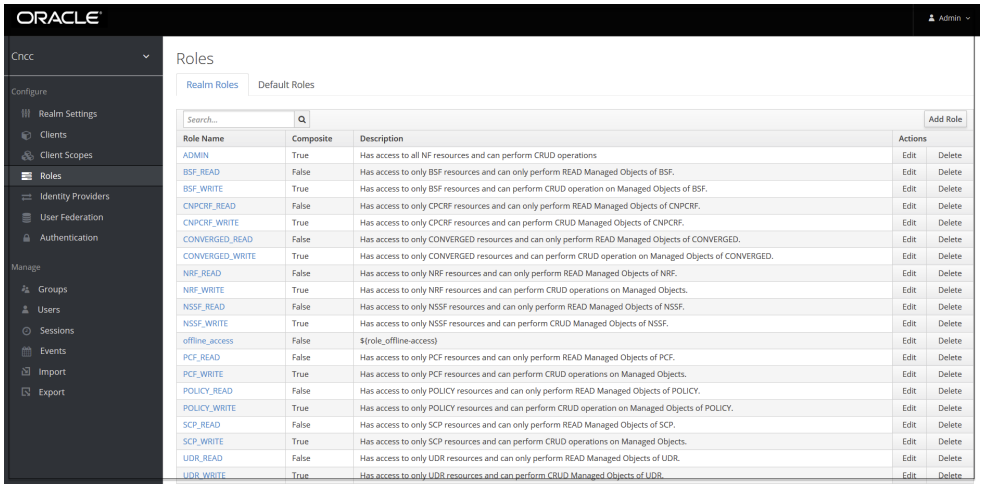


Viewing the Roles in CNC Console IAM

1. Select **Realm Settings** under **Cncc**.



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:

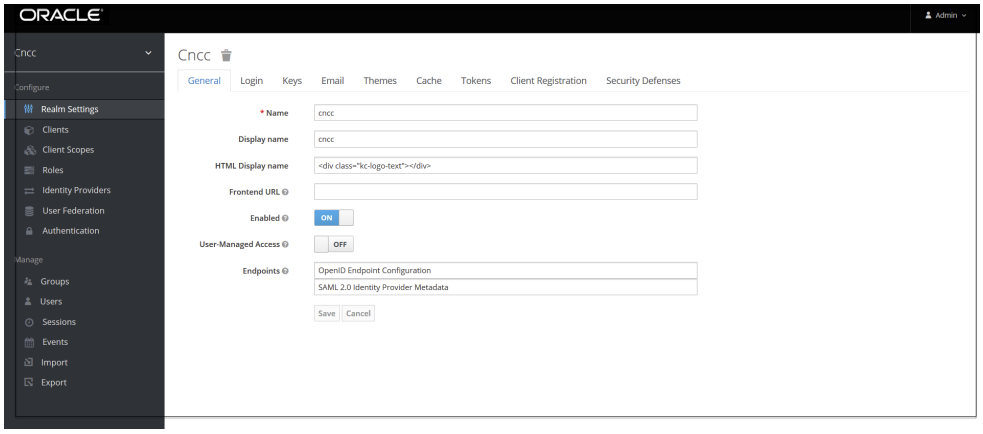
- Creating the users (not applicable if integrating with LDAP)
- Viewing the users
- Assigning the roles to the users (not applicable if integrating with LDAP)

> **Note:**
>
> For more details about the user roles refer APPENDIX.

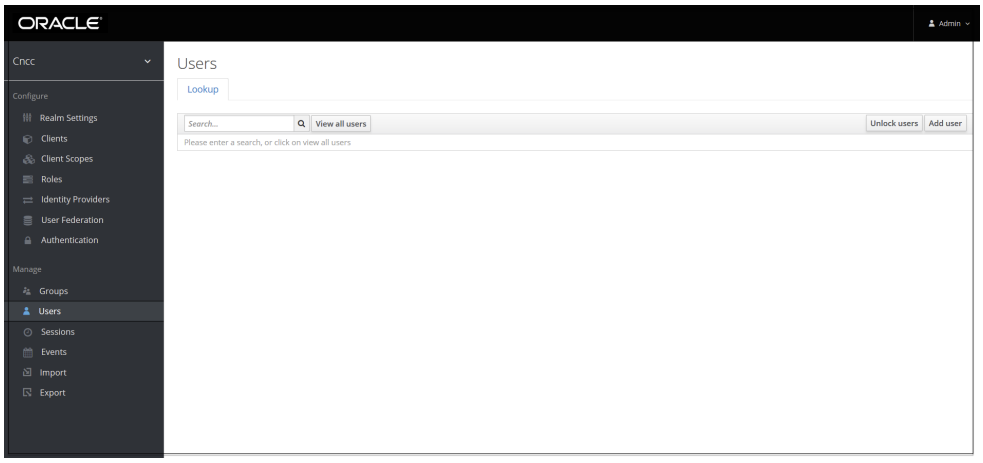
Creating the Users

> **Note:**
>
> This section is applicable only if CNC Console IAM is not integrated with LDAP.

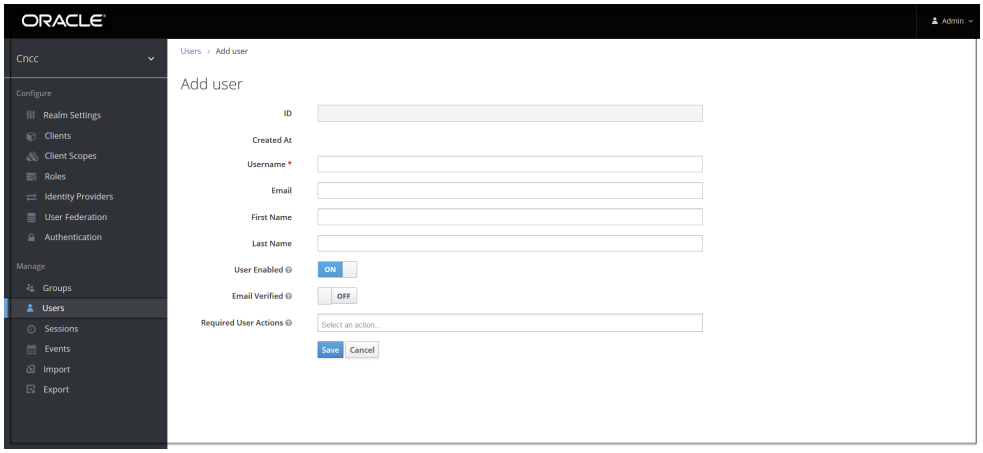
1. Click **Realm Settings** under **Cncc**.



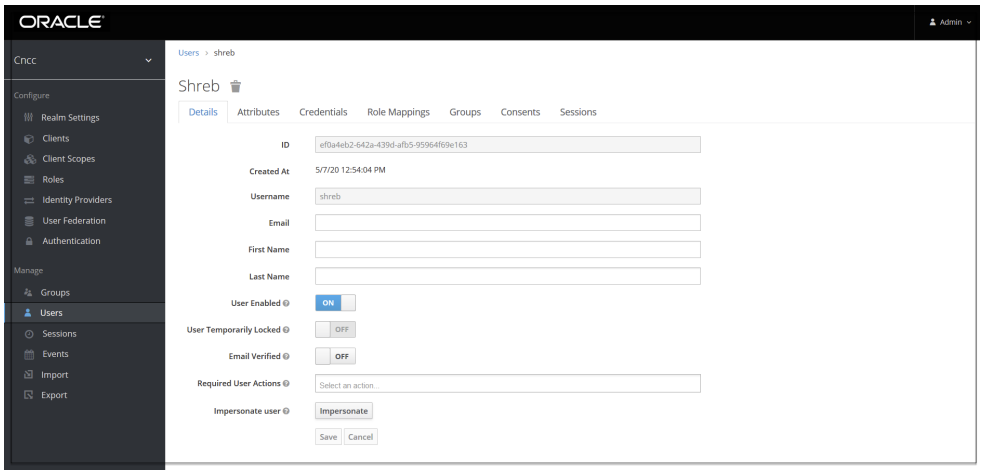
2. Select **Users** under **Manage** in the left pane and select **Add user** in the right pane.



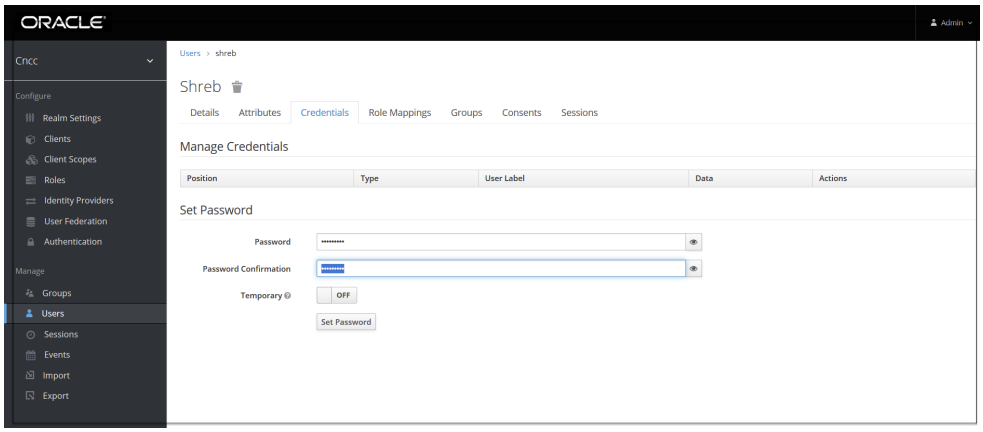
3. **Add user** Screen appears. Add the user details and click **Save.**



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



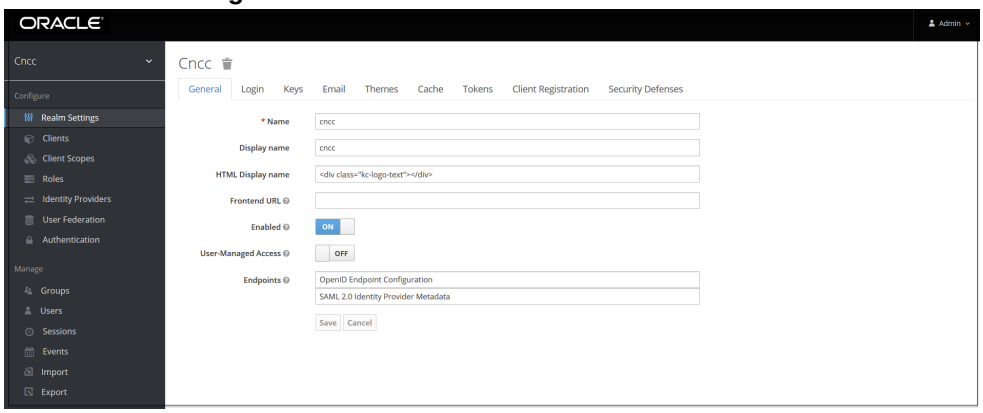
5. For setting the password for the user, Select **Users** under **Manage** in the left pane and select the **Credentials** tab in the right pane. Set the password for the user.



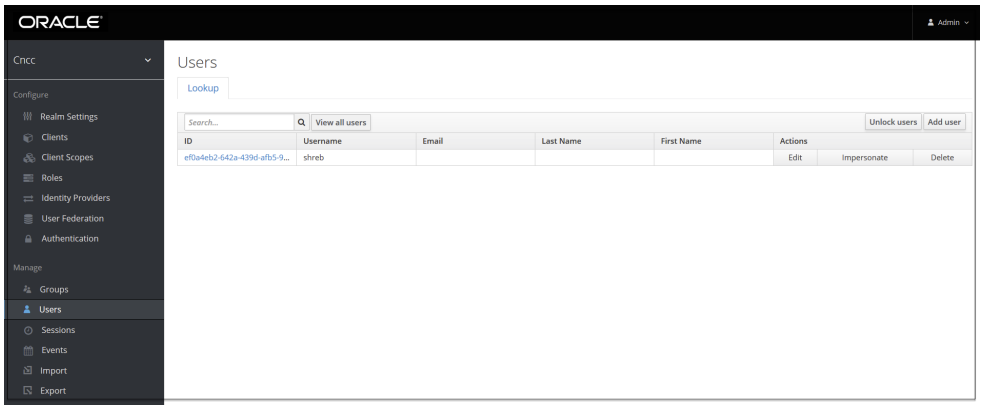
> **Note:**
>
> Setting the **Temporary** flag **ON** prompts the user to change the password while login for the first time to CNC Console Interface.

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.

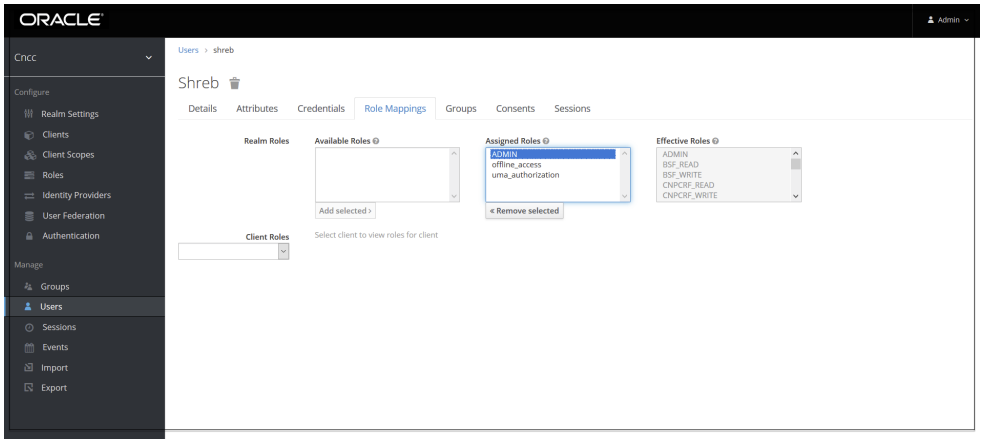


The list of users and their details appears in the right pane.

Assigning the Roles to User

> **Note:**
>
> This section is applicable only if CNC Console IAM is not integrated with LDAP.

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 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 ldif 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
```

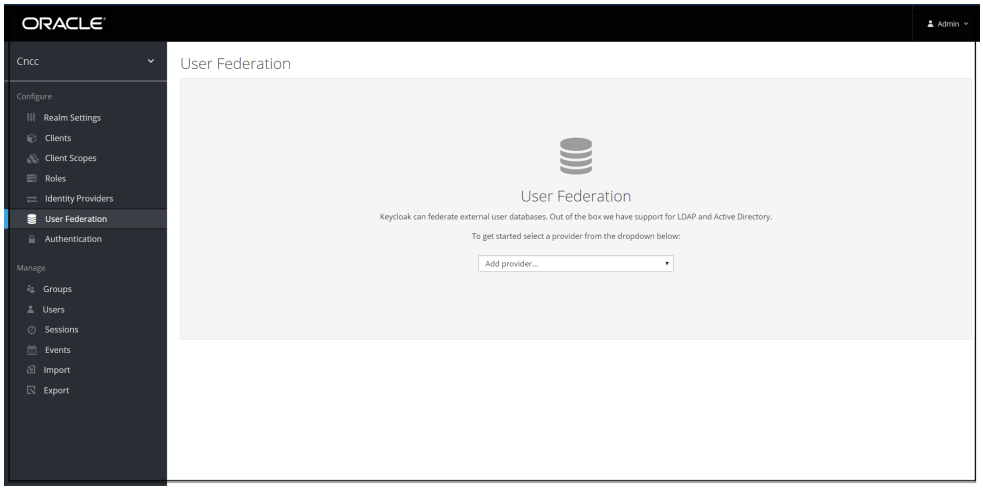
```
dn: ou=people,dc=oracle,dc=org
objectclass: top
objectclass: organizationalUnit
ou: people

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
```

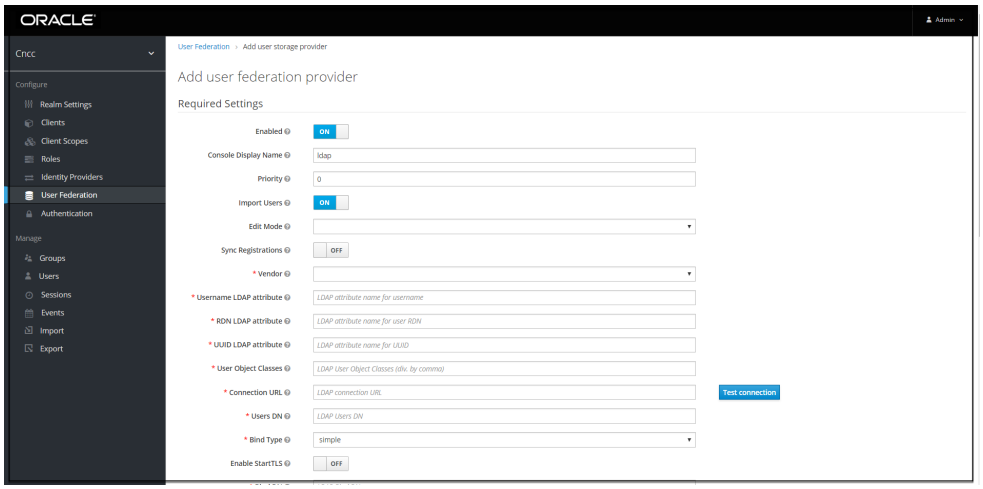
Setting up User Federation with CNC Console IAM (LDAP Server integration)

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.



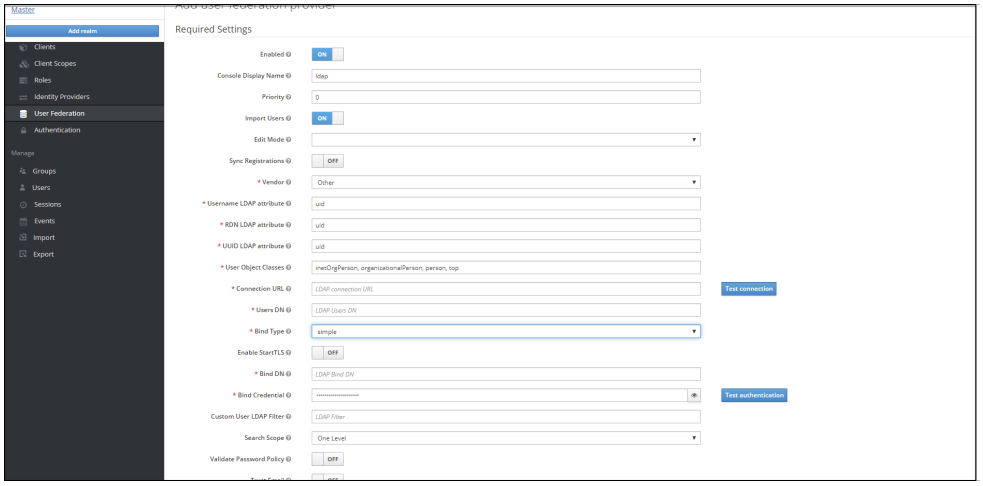
3. From the dropdown list in the **User federation** screen select **ldap**, the **Add user federation provider** screen appears.



4. Fill the following parameters:

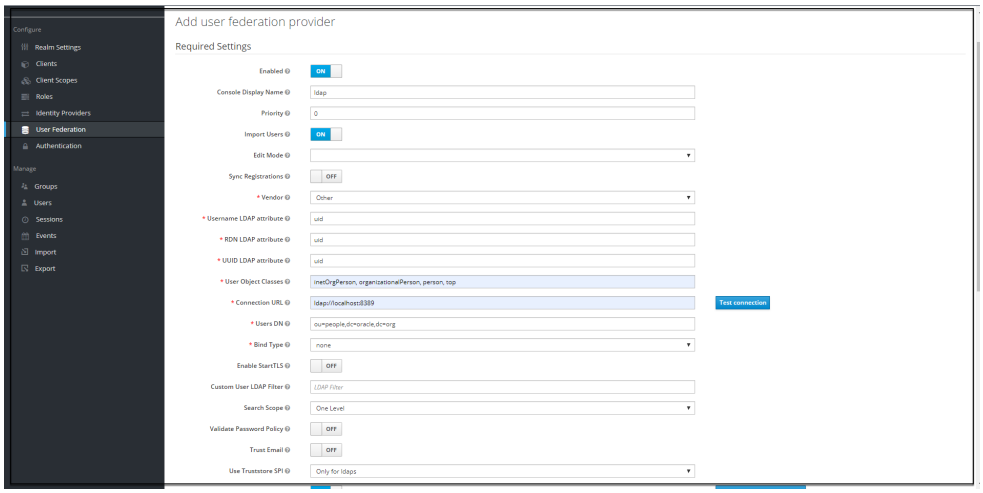
- **Console Display Name:** Enter the display name.
- **Vendor:** Enter the LDAP server provider name for the company.

> **Note:**
>
> This must usually fill the defaults for many of the fields. But in case you 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.

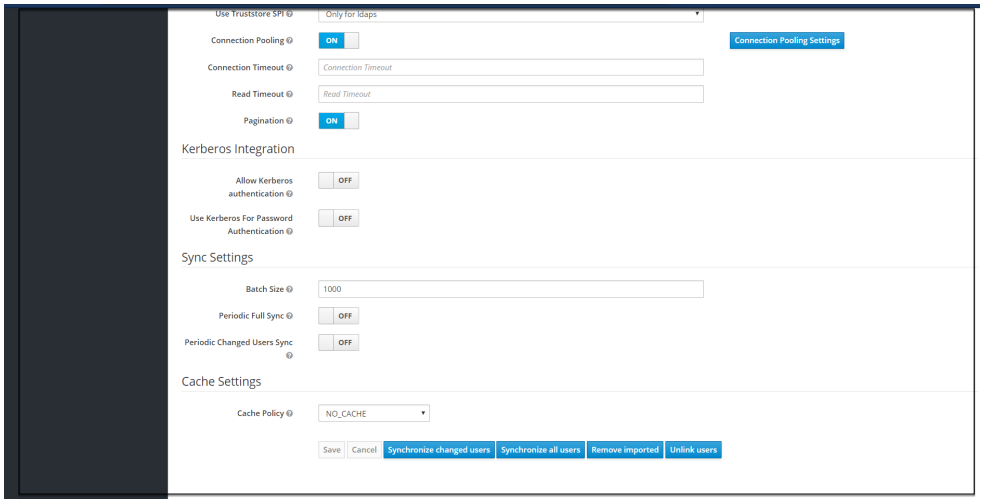


- Most companies have the **UUID LDAP attribute** value set as "entryUUID". If you don't have this field, than just use another unique identifier.
- Provide company LDAP server details.
- If LDAP is secured then provide the admin bind username and password else select **Bind-type** as "none".
- Click "Test Connection" and "Test Authentication".
- Set Cache policy as "NO_CACHE".

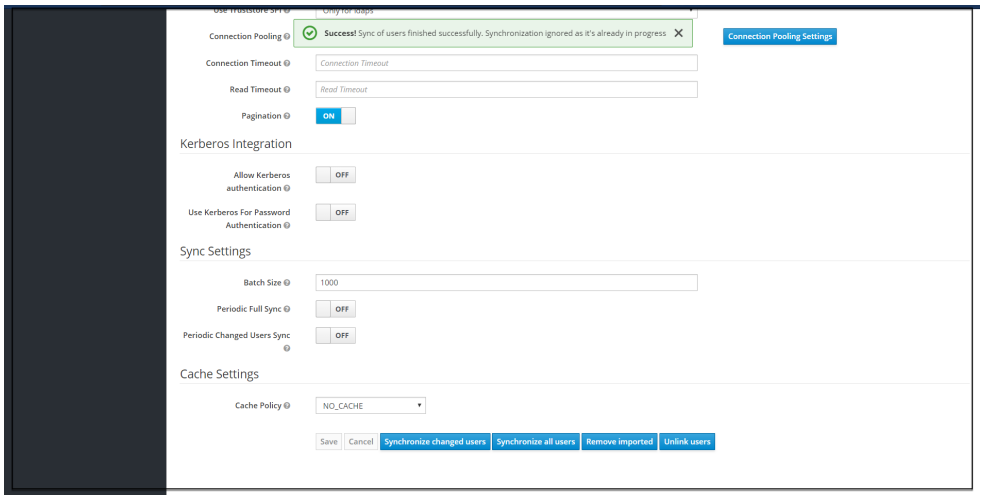
5. After filling the required fields, the screen appears as below. Click **Save**.



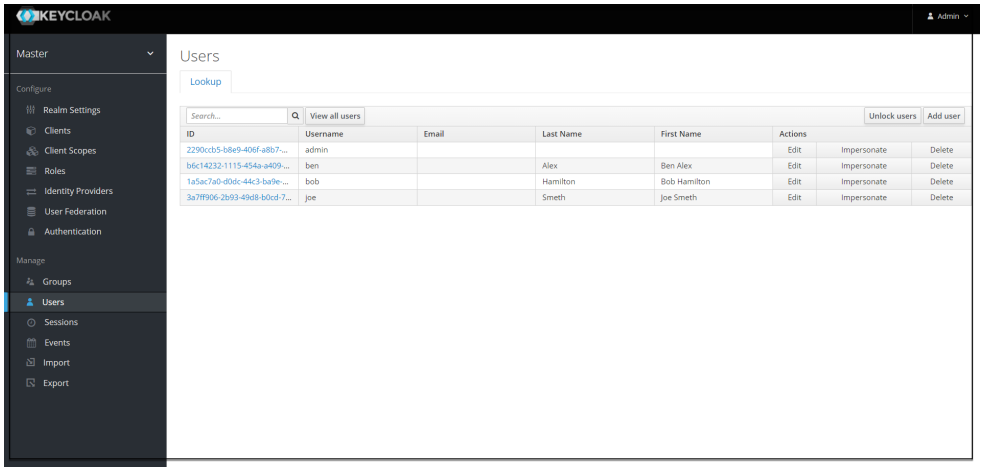
6. New buttons (**Synchronize changed users, Synchronize all users, Remove imported, Unlink users**) appears next to the **Save** and **Cancel**.



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.



> **Note:**
>
> The steps 8 and 9 are optional.

Group LDAP Mapper and Role Assignment

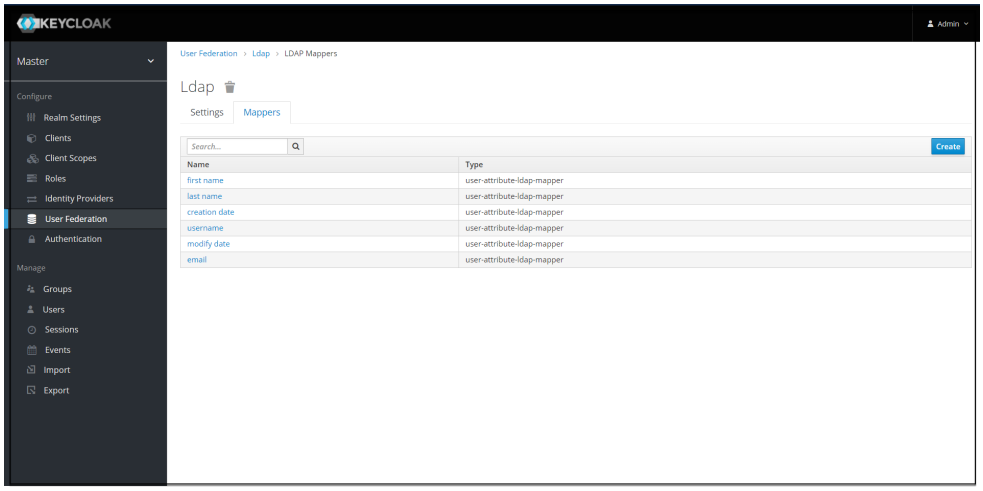
When an LDAP Federation provider is created, CNC Console-IAM provides a set of built-in mappers for this provider. User can change this set and create a new mapper or update/delete existing ones.

Group Mapper

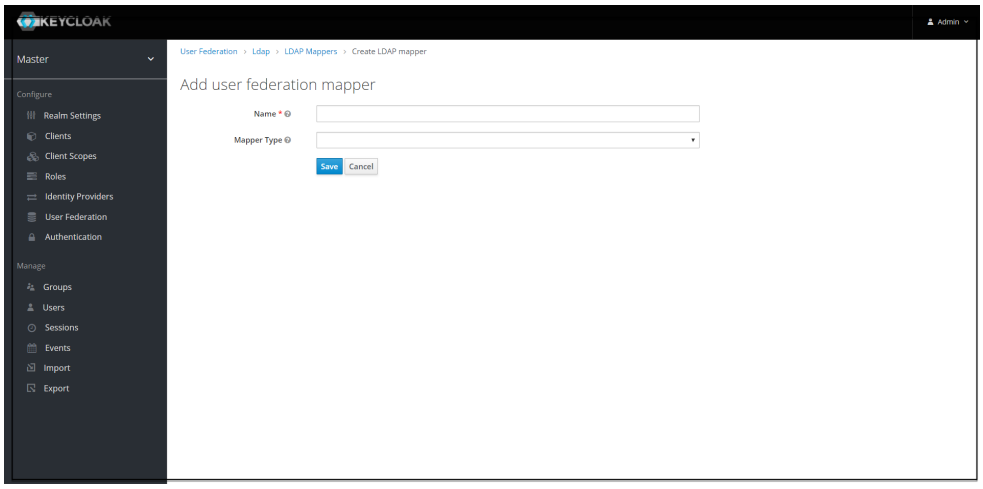
The Group Mapper allows you to configure group mappings from LDAP into cncc-iam group mappings. Group mapper can be used to map LDAP groups from a particular branch of an LDAP tree into groups in cncc-iam. It also propagates user-group mappings from LDAP into user-group mappings in cncc-iam.

To add Group-Mapper and assign roles:

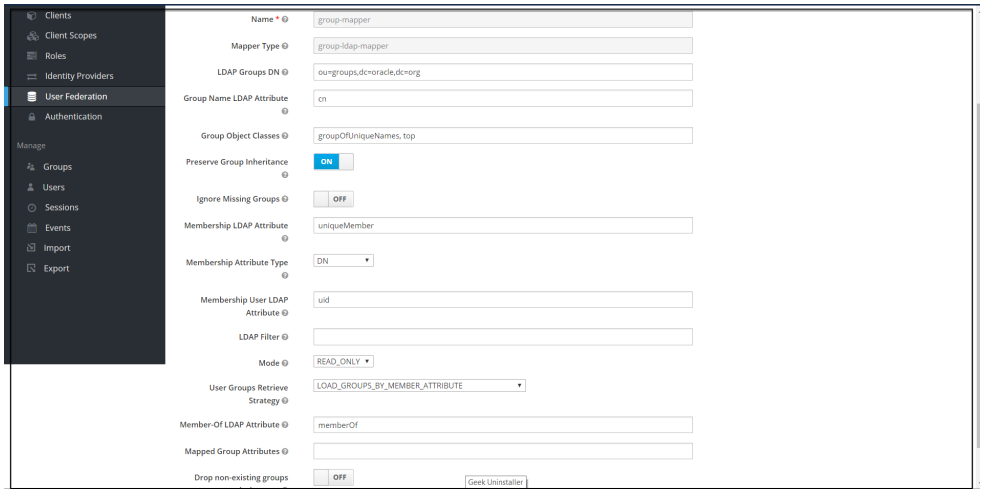
1. Click **Configure** and select **User Federation**. Click **ldap** (Console Display Name) and select the **Mappers** tab, and click **Create**.



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.**

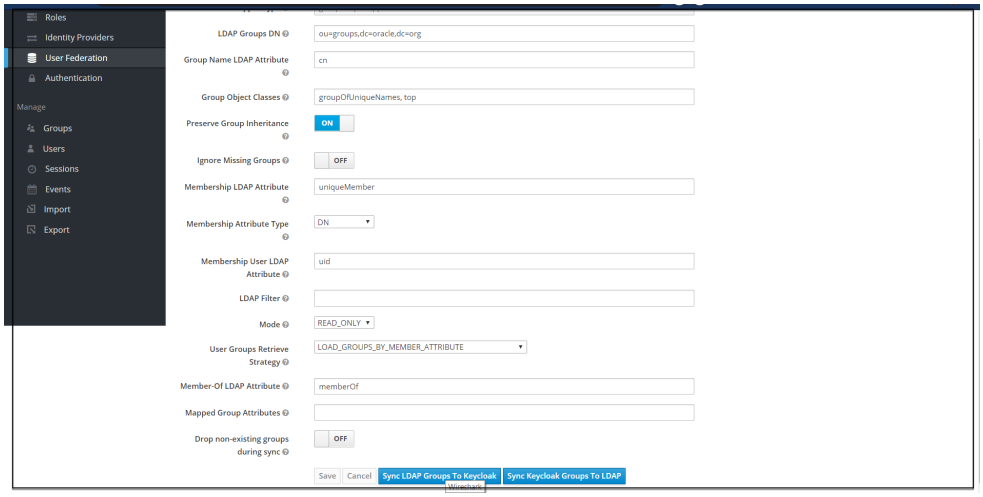


The following screen appears.

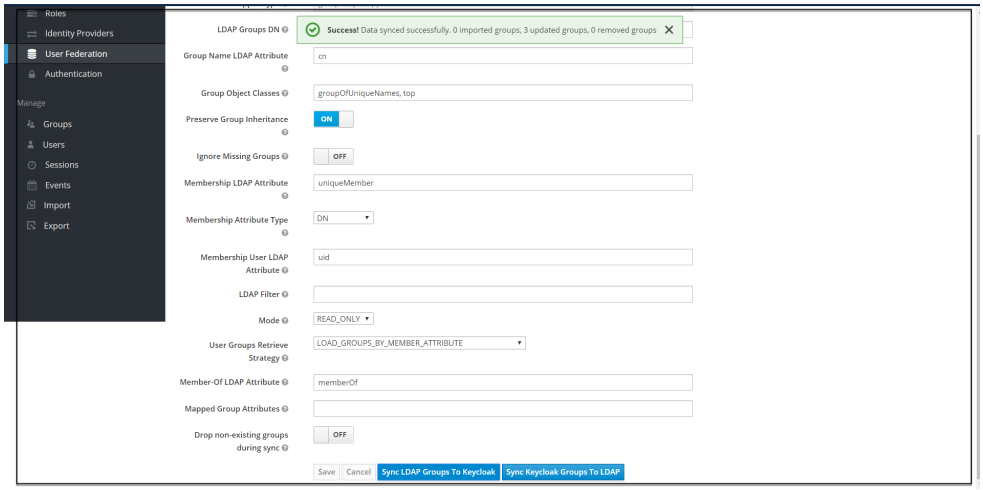


> **Note:**
>
> When selected, default values will be set by cncc-iam. But you need change some values based on your ldap records.

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

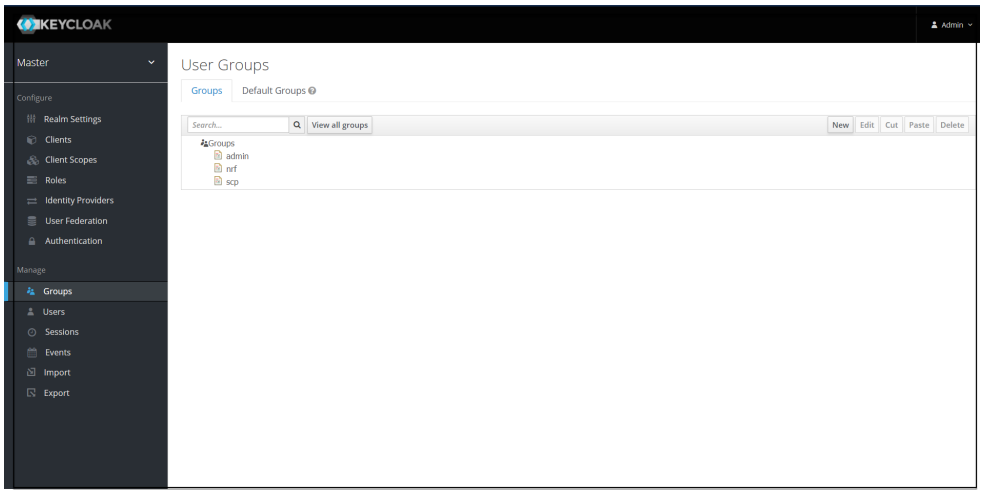


4. Click **Synchronize LDAP Groups to Keyclaok**. The success message appears with the number of groups imported and so on.

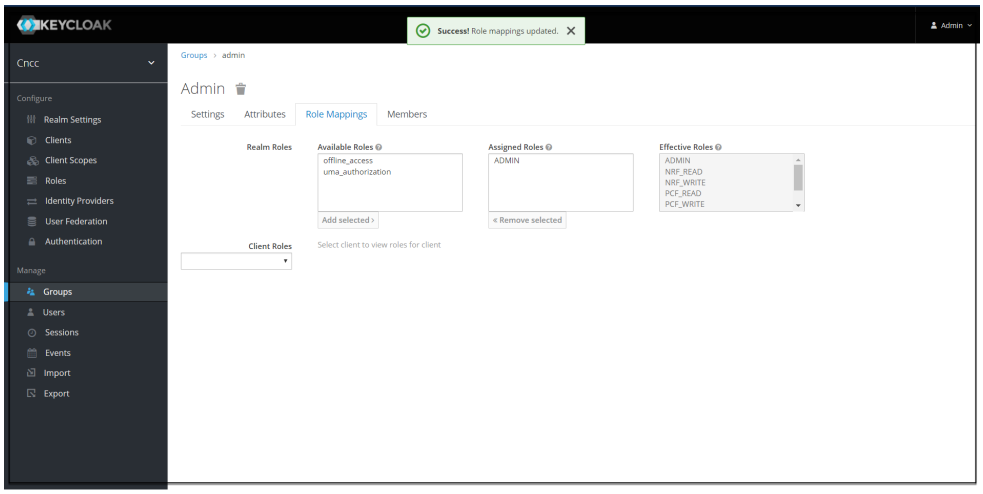


> **Note:**
>
> If this step fails then you might need to check to the trouble shooting section and look at cncc-iam logs in debug mode.

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



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



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

Note:

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

Note:

For more information about the user roles, refer APPENDIX.

6
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-url encoded* 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":
   "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJHS1N4WVhoWlExRVhrOVE
   ```

```
5RTR3STN4WG9LcHI2RW5yOFJCdGlMVndPV0JZIn0.eyJqdGkiOiIwMTQzYzNhZC1kNjE3LTQ
yNTYtYTA2My01NGYxNGI5MDQ5MWMiLCJleHAiOjE1ODQ3MDAyMzUsIm5iZiI6MCwiaWF0Ijo
xNTg0Njk5OTM1LCJpc3MiOiJodHRwOi8vMTAuNzUuMjEzLjYwOjMwMDI0L2NuY2MvYXV0aC9
yZWFsbXMvY25jYyIsImF1ZCI6ImFjY291bnQiLCJzdWIiOiIwMTJlNDI2OS02YzYwLTQzNWI
tYWExNC0yYWI3NmJjOWI1MjMiLCJ0eXAiOiJCZWFyZXIiLCJhenAiOiJhcGktZ2F0ZXdheSI
sImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6IjZjNDJkOTc4LTE0YWMtNDc5My1hMWU
zLTc4OWNmYmRiMmI3NCIsImFjciI6IjEiLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXMiOlsiUEN
GX1JFQUQiLCJQQ0ZfV1JJVEUiLCJTQ1BfUkVBRCIsIm9mZmxpbmVfYWNjZXNzIiwiTlJGX1d
SSVRFIiwiQURNSU4iLCJ1bWFfYXV0aG9yaXphdGlvbiIsIk5SRl9SRUFEIiwiU0NQX1dSSVR
FIl19LCJyZXNvdXJjZV9hY2Nlc3MiOnsiYWNjb3VudCI6eyJyb2xlcyI6WyJtYW5hZ2UtYWN
jb3VudCIsIm1hbmFnZS1hY2NvdW50LWxpbmtzIiwidmlldy1wcm9maWxlIl19fSwic2NvcGU
iOiJlbWFpbCBwcm9maWxlIiwiZW1haWxfdmVyaWZpZWQiOmZhbHNlLCJuYW1lIjoiU2hyZXl
hcyBCIiwicHJlZmVycmVkX3VzZXJuYW1lIjoic2hyZWIiLCJnaXZlbl9uYW1lIjoiU2hyZXl
hcyIsImZhbWlseV9uYW1lIjoiQiIsImVtYWlsIjoic2hyZXlhcy5iQG9yYWNsZS5jb20ifQ.
fXYyjmAbSSIFlLr2ZBEX2pfKrE_vr6Zbj8ta-
l_tKlv2gTX1J3ehScg_m30swpWU7UojuFkyc8CfNZL2Z9mcs7zbq_zA7ZTlaWA_AgmeoXWap
icX2wALT_YDU6Z3H7L9x1C1Ulp8aTBIBHPv2J-
zgkrFDtk83NeKunKEGlEZpp-9MGDLQ5a8QX6SAUo-
Fe6hNgF1vP0d7LCyjWvu6UvoeG_Fuxsi4xEVHcbSen8M3euеAt7xN7akhXZ_4PgWnxsWvQVq
tTzsY6O-
WyUjUiwtaTvpX0dPVVeeNDvWMY_0q0KvF_nnE3_wQtE8bu_LcCZYwDQJJTloj2PJ8y1WjO9l
2Q",
    "expires_in": 300,
    "refresh_expires_in": 1800,
    "refresh_token":
"eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICI3YTFlYjcyZi00MWE1LTR
kMTEtYjRmZS01NDZjZGU5NjY2MTUifQ.eyJqdGkiOiJmYjAwZTY2OC0xZTkyLTRlMTUtYTVl
MS1jZjgxNDFkMjllNDMiLCJleHAiOjE1ODQ3MDE3MzUsIm5iZiI6MCwiaWF0IjoxNTg0Njk5
OTM1LCJpc3MiOiJodHRwOi8vMTAuNzUuMjEzLjYwOjMwMDI0L2NuY2MvYXV0aC9yZWFsbXMv
Y25jYyIsImF1ZCI6Imh0dHA6Ly8xMC43NS4yMTMuNjA6MzAwMjQvY25jYy9hdXRoL3JlYWxt
cy9jbmNjIiwic3ViIjoiMDEyZTQyNjktNmM2MC00MzViLWFhMTQtMmFiNzZiYzliNTIzIiwi
dHlwIjoiUmVmcmVzaCIsImF6cCI6ImFwaS1nYXRld2F5IiwiYXV0aF90aW1lIjowLCJzZXNz
aW9uX3N0YXRlIjoiNmM0MmQ5NzgtMTRhYy00NzkzLWExZTMtNzg5Y2ZiZGIyYjc0IiwicmVh
bG1fYWNjZXNzIjp7InJvbGVzIjpbIlBDRl9SRUFEIiwiUENGX1dSSVRFIiwiU0NQX1JFQUQi
LCJvZmZsaW5lX2FjY2VzcyIsIk5SRl9XUklURSIsIkFETUlOIiwidW1hX2F1dGhvcml6YXRp
b24iLCJOUkZfUkVBRCIsIlNDUF9XUklURSJdfSwicmVzb3VyY2VfYWNjZXNzIjp7ImFjY291
bnQiOnsicm9sZXMiOlsibWFuYWdlLWFjY291bnQiLCJtYW5hZ2UtYWNjb3VudC1saW5rcyIs
InZpZXctcHJvZmlsZSJdfX0sInNjb3BlIjoiZW1haWwgcHJvZmlsZSJ9.l8w3j1gMNgblKSY
dvCmJQfg6yIfkdKnmFb5vKPF-ZIg",
    "token_type": "bearer",
    "not-before-policy": 0,
    "session_state": "6c42d978-14ac-4793-a1e3-789cfbdb2b74",
    "scope": "email profile"
}
```


5. The access token must be used in every request to a NF resource by placing it in the Authorization header. The Authorization header is given below:

```
GET : http://${cncc-core-ingress-external-ip}:${cncc-core-ingress-
service-port}/soothsayer/v1/canaryrelease
 headers: {
    'Authorization': 'Bearer' + access_token
}
```

6. Once the access_token has expired, it can be refreshed by sending a POST request to the same URL as above, but must have the refresh token instead of username and password. The format is given below:

```
'client_id': 'your_client_id',
'refresh_token': refresh_token_from_previous_request,
'grant_type': 'refresh_token'

Example:
'client_id': 'cncc',
'refresh_token':
'eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICI3YTFlYjcyZi00MWE1LTR
kMTEtYjRmZS01NDZjZGU5NjY2MTUifQ.eyJqdGkiOiJmYjAwZTY2OC0xZTkyLTRlMTUtYTVl
MS1jZjgxNDFkMjllNDMiLCJleHAiOjE1ODQ3MDE3MzUsIm5iZiI6MCwiaWF0IjoxNTg0Njk5
OTM1LCJpc3MiOiJodHRwOi8vMTAuNzUuMjEzLjYwOjMwMDI0L2NuY2MvYXV0aC9yZWFsbXMv
Y25jYyIsImF1ZCI6Imh0dHA6Ly8xMC43NS4yMTMuNjA6MzAwMjQvY25jYy9hdXRoL3JlYWxt
cy9jbmNjIiwic3ViIjoiMDEyZTQyNjktNmM2MC00MzViLWFhMTQtMmFiNzZiYzliNTIzIiwi
dHlwIjoiUmVmcmVzaCIsImF6cCI6ImFwaS1nYXRld2F5IiwiYXV0aF90aW1lIjowLCJzZXNz
aW9uX3N0YXRlIjoiNmM0MmQ5NzgtMTRhYy00NzkzLWExZTMtNzg5Y2ZiZGIyYjc0IiwicmVh
bG1fYWNjZXNzIjp7InJvbGVzIjpbIlBDRl9SRUFEIiwiUENGX1dSSVRFIiwiU0NQX1JFQUQi
LCJvZmZsaW5lX2FjY2VzcyIsIk5SRl9XUklURSIsIkFETUlOIiwidW1hX2F1dGhvcml6YXRp
b24iLCJOUkZfUkVBRCIsIlNDUF9XUklURSJdfSwicmVzb3VyY2VfYWNjZXNzIjp7ImFjY291
bnQiOnsicm9sZXMiOlsibWFuYWdlLWFjY291bnQiLCJtYW5hZ2UtYWNjb3VudC1saW5rcyIs
InZpZXctcHJvZmlsZSJdfX0sInNjb3BlIjoiZW1haWwgcHJvZmlsZSJ9.l8w3j1gMNgblKSY
dvCmJQfg6yIfkdKnmFb5vKPF-ZIg',
'grant_type': 'refresh_token'
```

7. In response user gets new **access_token** and **refresh_token**.

7
User Logs

The CNCC logs following user actions:

- **Login**: This log contains information about the users who are logged in to the system and their assigned roles.
- **NF Resource Access**: This log contains information about the name of the users who are accessing the resource and and their assigned roles.
- **Logout**: This log contains information about the users who are logged out from the system.

Log Details

Table 7-1 Log Details

Name	Description	Example
thread	Name of the thread	"thread": "reactor-http-epoll-1"
level	Level of the log. It can be: Log level (INFO, WARN, DEBUG, TRACE)	"level": "INFO"
loggerName	Name of the class that generated the log	"loggerName": "ocpm.cne.gateway.cncc.config.CustomAuthenticationSuccessHandler"
message	Information about the event	"message": "Login successful -- UserName: jon"
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-core-ingress-gateway-77df795fb5-wv2sb"
contextMap	Information about host-name and transaction-id	"contextMap": { "hostname": "cncc-core-ingress-gateway-77df795fb5-wv2sb", "ingressTxId": "ingress-tx-1460885598"}

Log Format

The format of log is as follows:

```
{
     "thread": <threadId>,
     "level": <log_level>,
```

```
     "loggerName": <name_of_the_class>,
     "message": <message>,
     "instant": <timestamp_in_miliseconds>,
     "threadId": <threadId>,
     "threadPriority": <threadPriority>,
     "pod": <name_of_the_pod>,
     "processId": <processId>,
     "contextMap": <context_map>
}
```

Log Examples

User Login Example:

```
{
    "thread": "reactor-http-epoll-1",
    "level": "INFO",
    "loggerName":
"ocpm.cne.gateway.cncc.config.CustomAuthenticationSuccessHandler",
    "message": "Login successful -- UserName: shreb, Roles: [\"PCF_WRITE\",
\"CONVERGED_READ\",\"CONVERGED_WRITE\",\"UDR_READ\",\"POLICY_READ\",
\"PCF_READ\",\"SCP_READ\",\"UDR_WRITE\",\"CNPCRF_WRITE\",\"NRF_WRITE\",
\"ADMIN\",\"POLICY_WRITE\",\"CNPCRF_READ\",\"NRF_READ\",\"SCP_WRITE\"]",
    "endOfBatch": false,
    "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLogger",
    "instant": {
        "epochSecond": 1590045098,
        "nanoOfSecond": 143345000
    },
    "contextMap": {},
    "threadId": 43,
    "threadPriority": 5,
    "processId": "1",
    "instanceType": "prod"
}
{
    "thread": "reactor-http-epoll-1",
    "level": "INFO",
    "loggerName":
"ocpm.cne.gateway.cncc.config.CustomAuthenticationSuccessHandler",
    "message": "Session timeoutPT30M , Session ID :
c9fdf0d5-0676-4489-9685-3122f2e77568",
    "endOfBatch": false,
    "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLogger",
    "instant": {
        "epochSecond": 1590045356,
        "nanoOfSecond": 575067000
    },
    "contextMap": {
        "hostname": "cncc-core-ingress-gateway-77df795fb5-wv2sb",
        "ingressTxId": "ingress-tx-1460885598"
    },
    "threadId": 43,
    "pod": "cncc-core-ingress-gateway-77df795fb5-wv2sb",
    "threadPriority": 5,
```

```
    "processId": "1",
    "instanceType": "prod",
    "ingressTxId": "ingress-tx-1460885598"
}
```


User NF Resource Access Example:


```
{
    "thread": "reactor-http-epoll-2",
    "level": "INFO",
    "loggerName":
"ocpm.cne.gateway.cncc.config.RolesBasedAuthorizationDecision",
    "message": "User Authorization Details -- UserName: shreb, Roles:
[\"PCF_WRITE\",\"CONVERGED_READ\",\"CONVERGED_WRITE\",\"UDR_READ\",
\"POLICY_READ\",\"PCF_READ\",\"SCP_READ\",\"UDR_WRITE\",\"CNPCRF_WRITE\",
\"NRF_WRITE\",\"ADMIN\",\"POLICY_WRITE\",\"CNPCRF_READ\",\"NRF_READ\",
\"SCP_WRITE\"]",
    "endOfBatch": false,
    "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLogger",
    "instant": {
        "epochSecond": 1590045388,
        "nanoOfSecond": 339789000
    },
    "contextMap": {
        "hostname": "cncc-core-ingress-gateway-77df795fb5-wv2sb",
        "ingressTxId": "ingress-tx-175723908"
    },
    "threadId": 56,
    "threadPriority": 5,
    "pod": "cncc-core-ingress-gateway-77df795fb5-wv2sb",
    "processId": "1",
    "instanceType": "prod",
    "ingressTxId": "ingress-tx-175723908"
}
```


User Logout Example:


```
{
    "thread": "reactor-http-epoll-2",
    "level": "INFO",
    "loggerName":
"ocpm.cne.gateway.cncc.config.CustomServerLogoutSuccessHandler",
    "message": "Logout successful -- UserName: shreb",
    "endOfBatch": false,
    "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLogger",
    "instant": {
        "epochSecond": 1590045260,
        "nanoOfSecond": 379367000
    },
    "contextMap": {
        "hostname": "cncc-core-ingress-gateway-77df795fb5-wv2sb",
        "ingressTxId": "ingress-tx-1989016497"
    },
    "threadId": 56,
```

```
    "threadPriority": 5,
    "pod": "cncc-core-ingress-gateway-77df795fb5-wv2sb",
    "processId": "1",
    "instanceType": "prod",
    "ingressTxId": "ingress-tx-1989016497"
}
```

Accessing the Logs

CNCC application logs can be accessed using any one of following options:

Using Command:

- Execute the following commad to view the logs of a running CNCC application pod:

`kubectl logs -f -n <cncc_namespace> <pod_name> -c <container_name>`
Example: `kubectl logs -f -n cncc cncc-core-ingress-gateway-77df795fb5-wv2sb -c ingress-gateway`

Using Logging Framework:

- View logs using cloud native supported logging framework:

  Example : EFK (Elasticsearch, Fluentd and Kibana) can be used with CNCC to view the logs.

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.

B

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 PCF and SCP NFs then, PCF_READ, PCF_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 PCF_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 PCF_WRITE then the user can read/write/update/delete any MOs configurations within the NF.