Oracle® Communications

Cloud Native Core Console User's Guide

Release 1.2.1
F32764-02
August 2020

Oracle Communications Cloud Native Core Console User's Guide, Release 1.2.1

F32764-02

Contents

My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

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 document is republished for **Release 1.2.1.** No updates have been incorporated in this document for this release.

New and Updated features in Release 1.2:

- SAML SSO is integrated with CNC Console IAM
- CNPCRF and PCF are converged to POLICY
- UDR, SCP and NRF GUIs are enhanced
- Security Logs and User Activity Logs are enhanced

List of Tables

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

> **Note:**
>
> Currently, CNCC 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
NRF	Network Repository Function
OSDC	Oracle Software Delivery Cloud
SCP	Service Communication Proxy
AMF	Access and Mobility Management Function
IAM	Identity Access Management
UDR	Unified Data Repository
UE	User Equipment
LDAP	Lightweight Directory Access Protocol
HTTPS	Hypertext Transfer Protocol Secure

2
CNC Console

This section provides information about CNC Console.

The CNC Console has two modules:

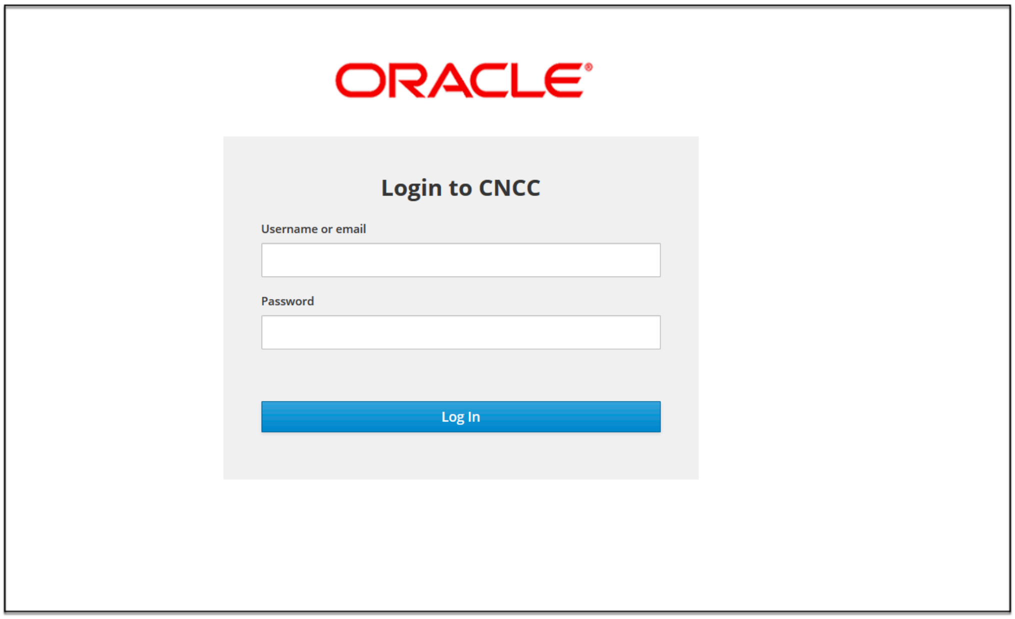
- **CNC Console Core (CNCC Core)**: CNCC Core module includes the GUI aspects of the interface. The integration of all the supported NFs are included in this module.
- **CNC Identity Access Management (CNC IAM)**: CNC IAM module includes the authentication and authorization aspects of the interface. 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 cncc-iam-ingress-ip and *<port number> is cncc-iam-ingress-port*

3. Enter the valid credentials.

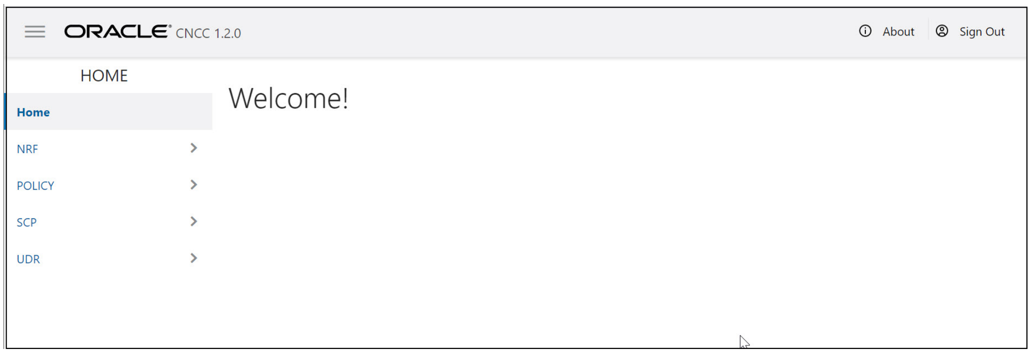
4. Click **Log In**. The Welcome Screen of CNC Console interface appears.

> **Note:**
>
> To set up CNCC IAM, refer to **Setting up CNC Console IAM** section

Working on CNC Console

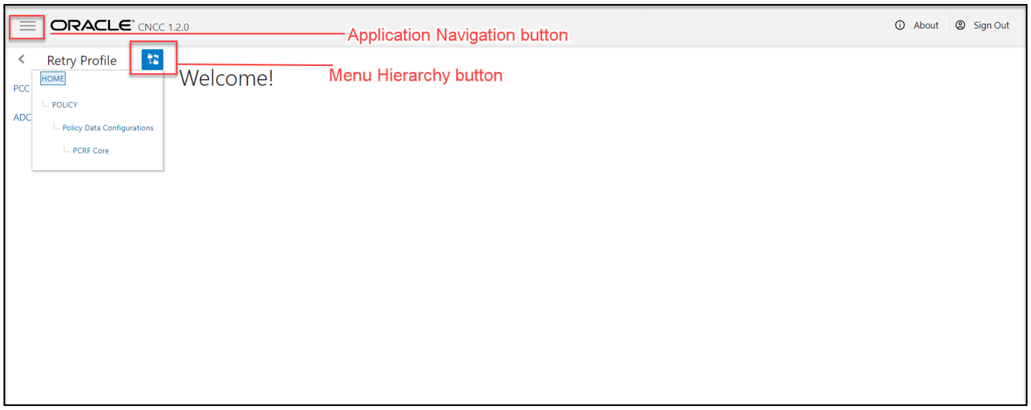
GUI Details

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



Welcome!

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 .



Welcome!

Application Navigation button

Menu Hierarchy button

Note:

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

Types of User Interface Screens for NF Configurations

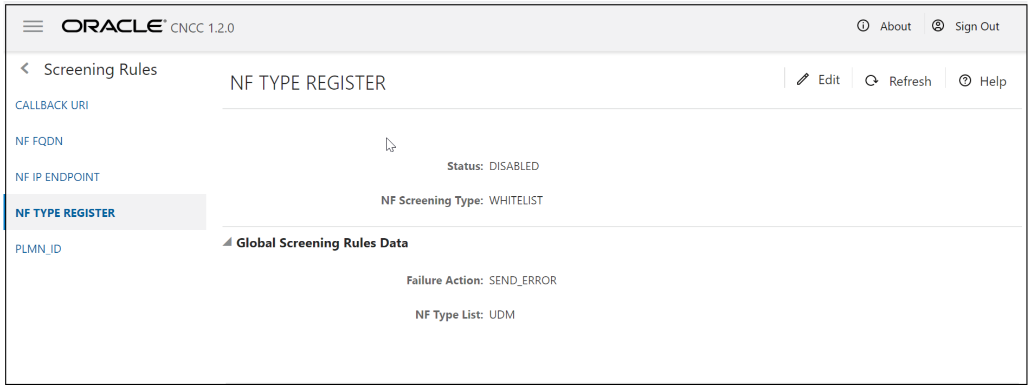
CNC Console Screens

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



NF TYPE REGISTER

Status: DISABLED

NF Screening Type: WHITELIST

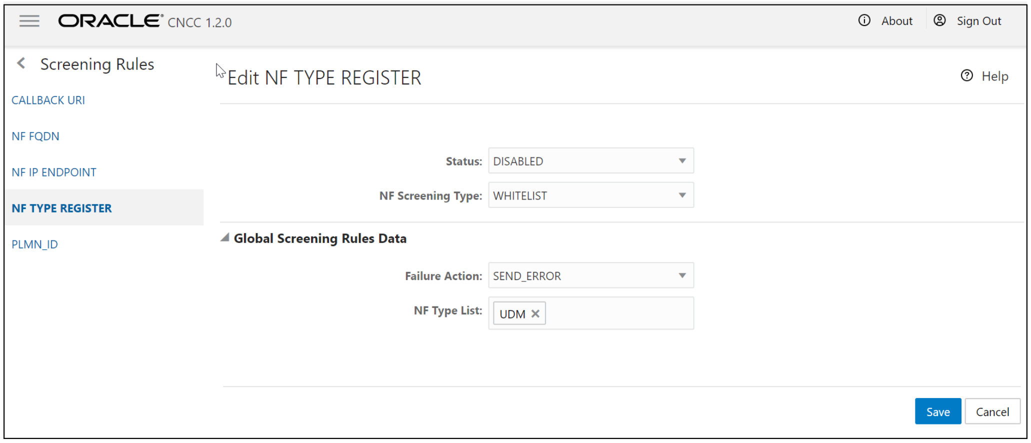
Global Screening Rules Data

Failure Action: SEND_ERROR

NF Type List: UDM

Service Screen - Edit

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



Edit NF TYPE REGISTER

Status: DISABLED

NF Screening Type: WHITELIST

Global Screening Rules Data

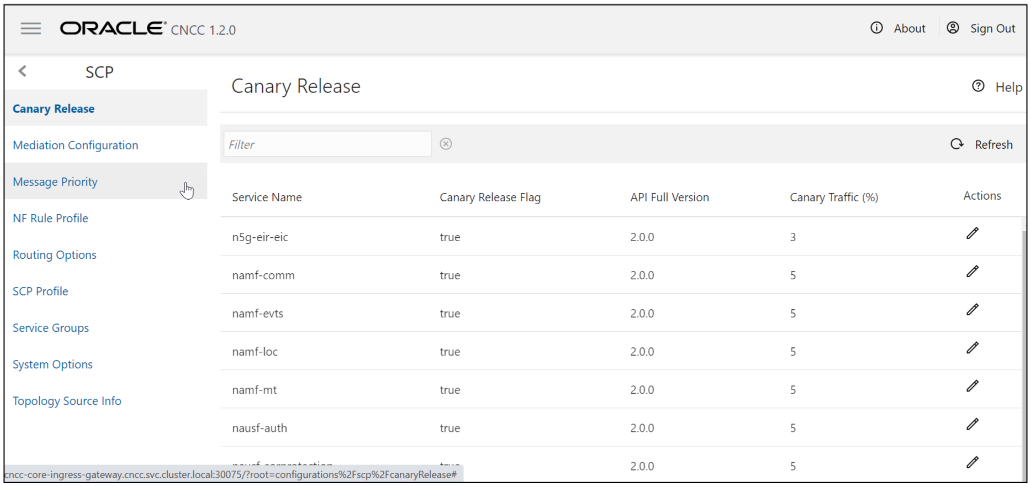
Failure Action: SEND_ERROR

NF Type List: UDM ×

Configurations Screen

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

Examples for the Configuration Screen:



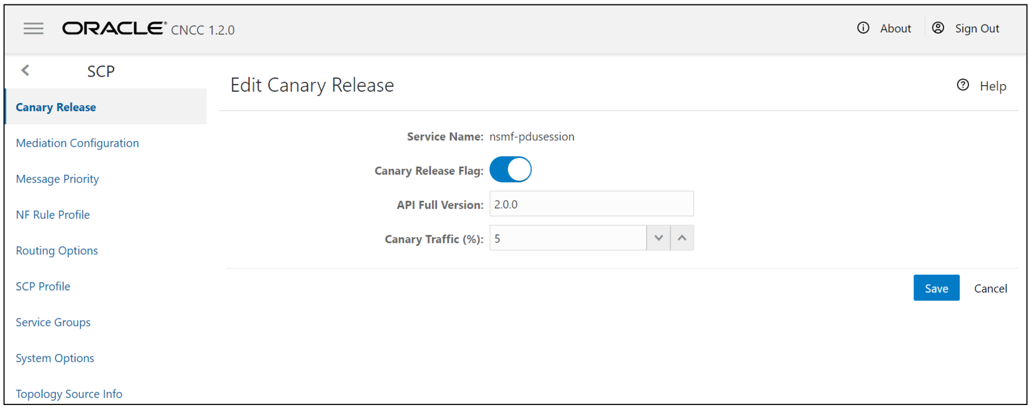
Canary Release

Service Name	Canary Release Flag	API Full Version	Canary Traffic (%)	Actions
n5g-eir-eic	true	2.0.0	3	✎
namf-comm	true	2.0.0	5	✎
namf-evts	true	2.0.0	5	✎
namf-loc	true	2.0.0	5	✎
namf-mt	true	2.0.0	5	✎
nausf-auth	true	2.0.0	5	✎
nausf-sorprotection	true	2.0.0	5	✎

cncc-core-ingress-gateway.cncc.svc.cluster.local:30075/?root=configurations%2Fscp%2FcanaryRelease#

Configurations Screen- Edit

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



Edit Canary Release

Service Name: nsmf-pdusession

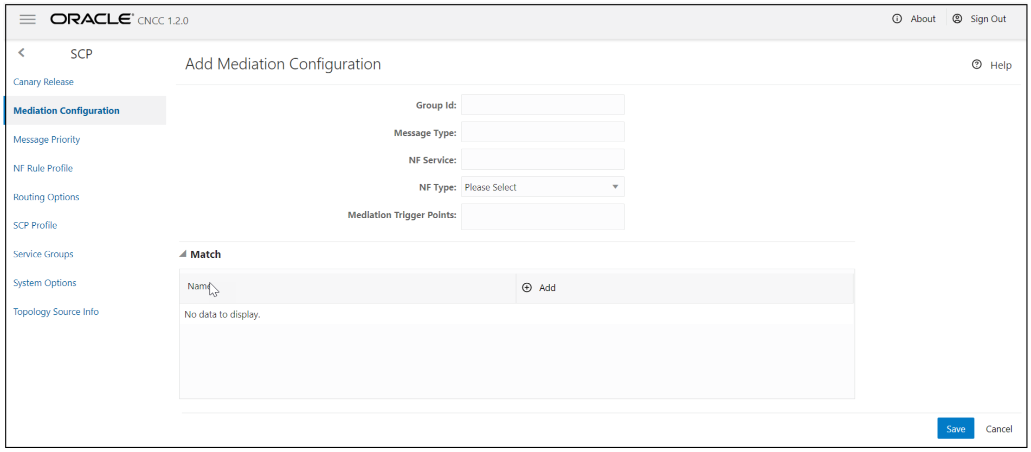
Canary Release Flag: ●

API Full Version: 2.0.0

Canary Traffic (%): 5

Configurations Screen - Add

The Configurations screen with **Add** enabled.



Add Mediation Configuration

Group Id:

Message Type:

NF Service:

NF Type: Please Select

Mediation Trigger Points:

Match

Name
No data to display.

3 Configuring Network Functions

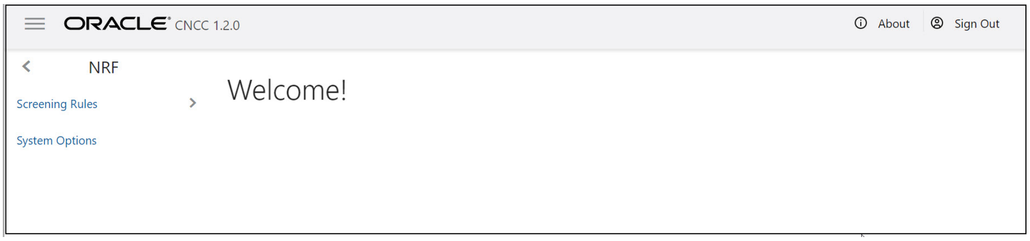
Network Repository Functions (NRF)

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

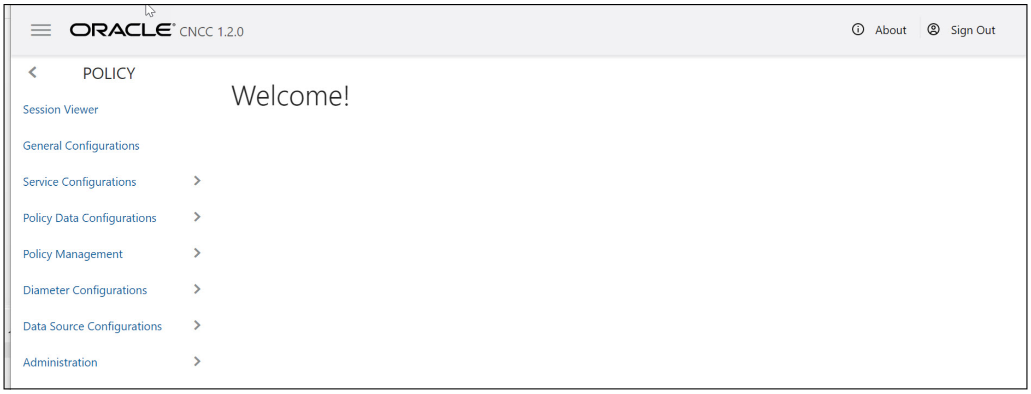
Oracle Communications Cloud Native Core Policy (CNC Policy) solution provides a standard policy design experience and ultimately consistent end-user experience. The Converged policy solution supports both 4G and 5G networks. In addition, the overlap in functionality between PCF and PCRF (Example: need 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 Moblity, 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**.

18. 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.**

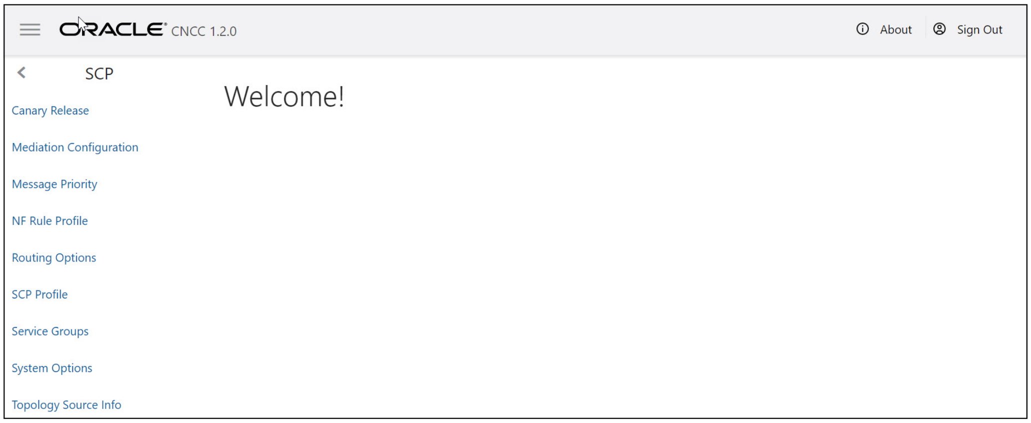
Service Communication Proxy (SCP)

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:



1. 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**.
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.

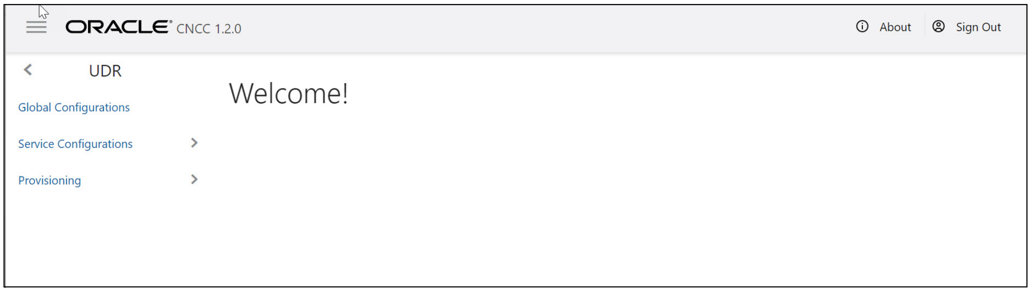
Oracle 5G UDR is implemented as cloud native function and it offers a unified database for storing application, subscription, authentication, service authorization, policy data, session binding and application state information. It exposes a HTTP2 based RESTful API for Network Functions and provisioning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

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 1.2.0:

NF	NF Version
SCP	1.7.0
NRF	1.7.0
Policy	1.7.0
UDR	1.7.0

4
Setting up CNC Console IAM

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

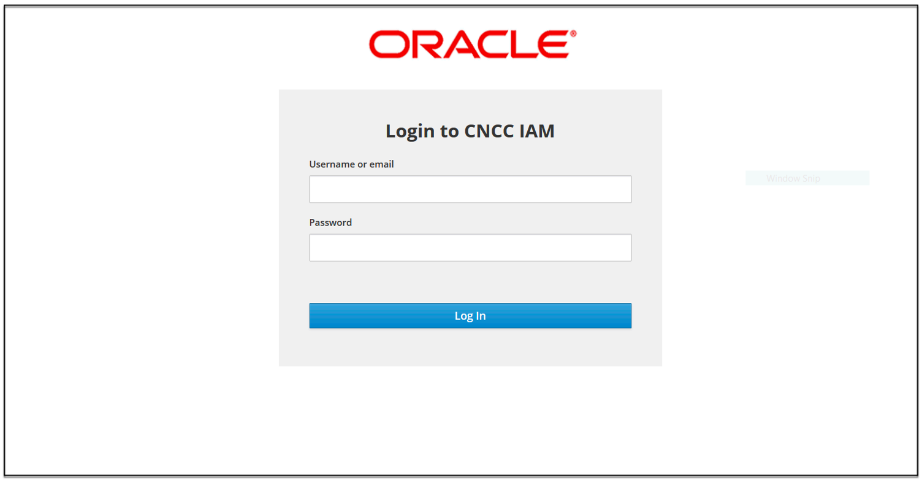
Setting up the CNCC Redirection URL

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/*
   ```

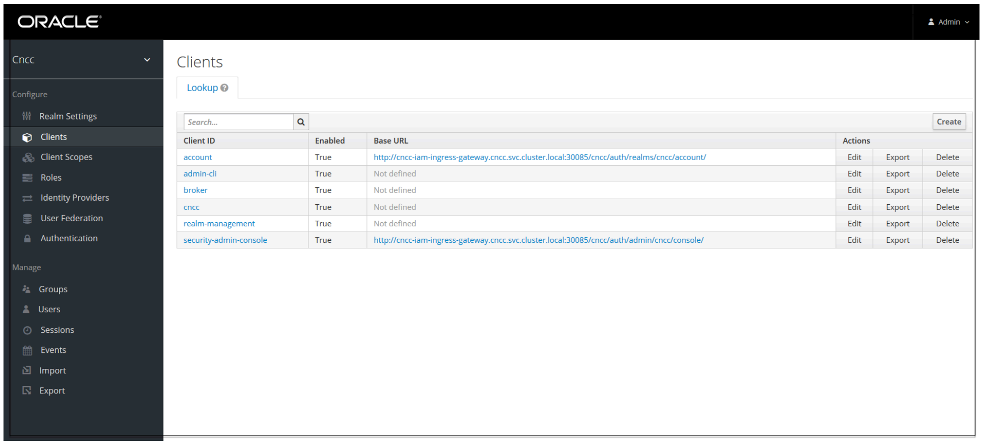


Login to CNCC IAM

Username or email

Password

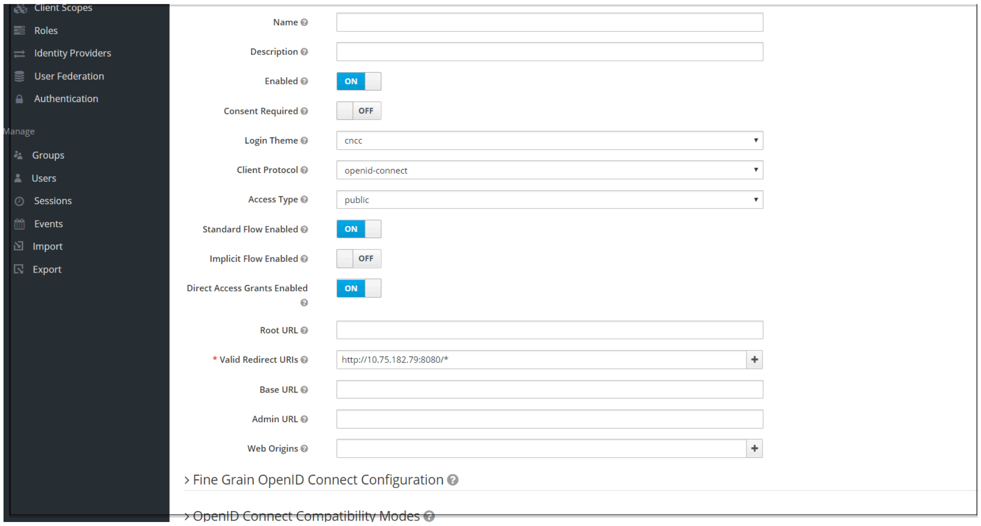
2. Go to **Clients** and select **Cncc**.



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

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

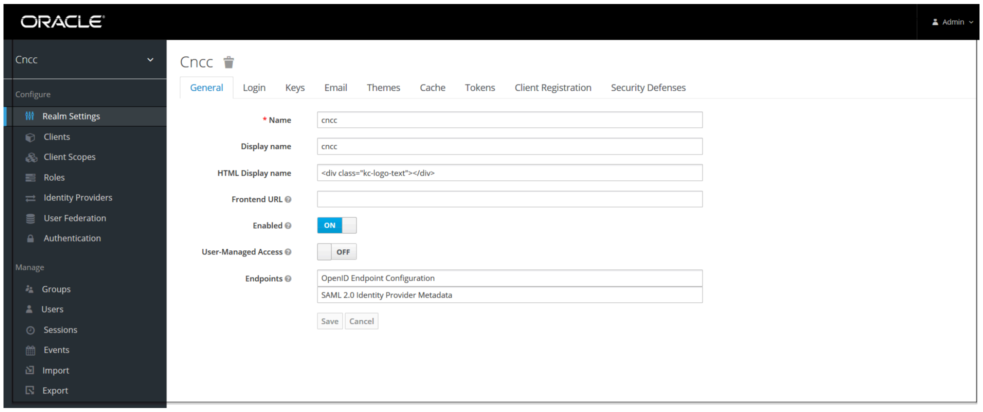
```
<scheme>://<cncc-core-ingress-extrenal-ip>:<cncc-core-ingress-
service-port>/*
 Example:  http://10.75.182.79:8080/*
```



Name	
Description	
Enabled	ON
Consent Required	OFF
Login Theme	cncc
Client Protocol	openid-connect
Access Type	public
Standard Flow Enabled	ON
Implicit Flow Enabled	OFF
Direct Access Grants Enabled	ON
Root URL	
* Valid Redirect URIs	http://10.75.182.79:8080/*
Base URL	
Admin URL	
Web Origins	

Viewing the Roles in CNC Console IAM

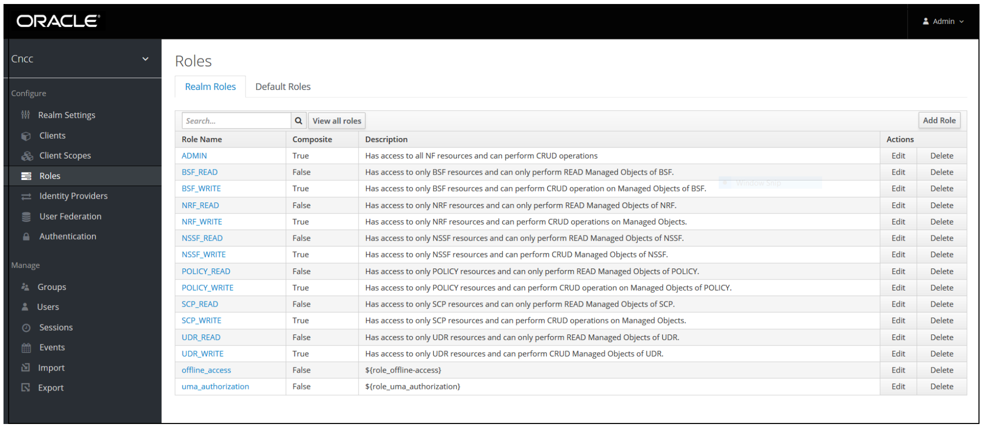
1. Select **Realm Settings** under **Cncc**.



Cncc

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

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



Roles

Role Name	Composite	Description	Actions	
ADMIN	True	Has access to all NF resources and can perform CRUD operations	Edit	Delete
BSF_READ	False	Has access to only BSF resources and can only perform READ Managed Objects of BSF.	Edit	Delete
BSF_WRITE	True	Has access to only BSF resources and can perform CRUD operation on Managed Objects of BSF.	Edit	Delete
NRF_READ	False	Has access to only NRF resources and can only perform READ Managed Objects of NRF.	Edit	Delete
NRF_WRITE	True	Has access to only NRF resources and can perform CRUD operations on Managed Objects.	Edit	Delete
NSSF_READ	False	Has access to only NSSF resources and can only perform READ Managed Objects of NSSF.	Edit	Delete
NSSF_WRITE	True	Has access to only NSSF resources and can perform CRUD Managed Objects of NSSF.	Edit	Delete
POLICY_READ	False	Has access to only POLICY resources and can only perform READ Managed Objects of POLICY.	Edit	Delete
POLICY_WRITE	True	Has access to only POLICY resources and can perform CRUD operation on Managed Objects of POLICY.	Edit	Delete
SCP_READ	False	Has access to only SCP resources and can only perform READ Managed Objects of SCP.	Edit	Delete
SCP_WRITE	True	Has access to only SCP resources and can perform CRUD operations on Managed Objects.	Edit	Delete
UDR_READ	False	Has access to only UDR resources and can only perform READ Managed Objects of UDR.	Edit	Delete
UDR_WRITE	True	Has access to only UDR resources and can perform CRUD Managed Objects of UDR.	Edit	Delete
offline_access	False	${role_offline-access}	Edit	Delete
uma_authorization	False	${role_uma_authorization}	Edit	Delete

Users and Roles in CNC Console IAM

This section includes:

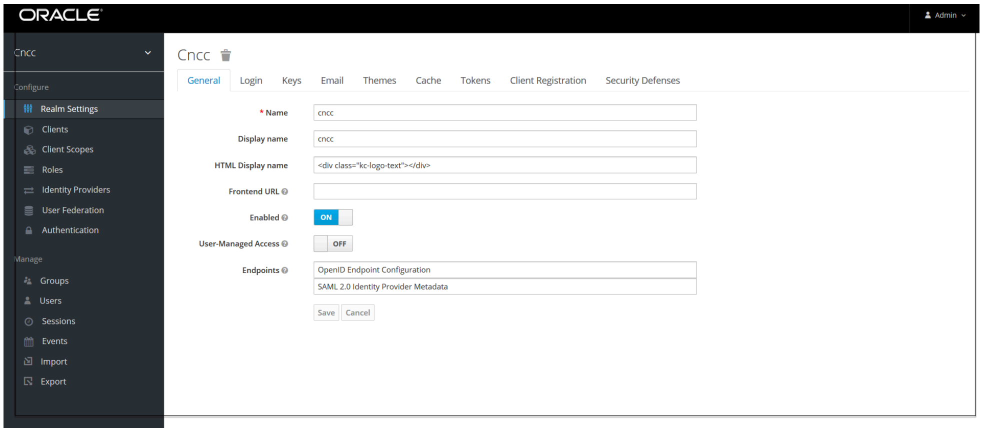
- Creating the users (not applicable if integrating with LDAP)
- Viewing the users
- Assigning the roles to the users (not applicable if integrating with LDAP)

> **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**.



* Name: cncc

Display name: cncc

HTML Display name: <div class="kc-logo-text"></div>

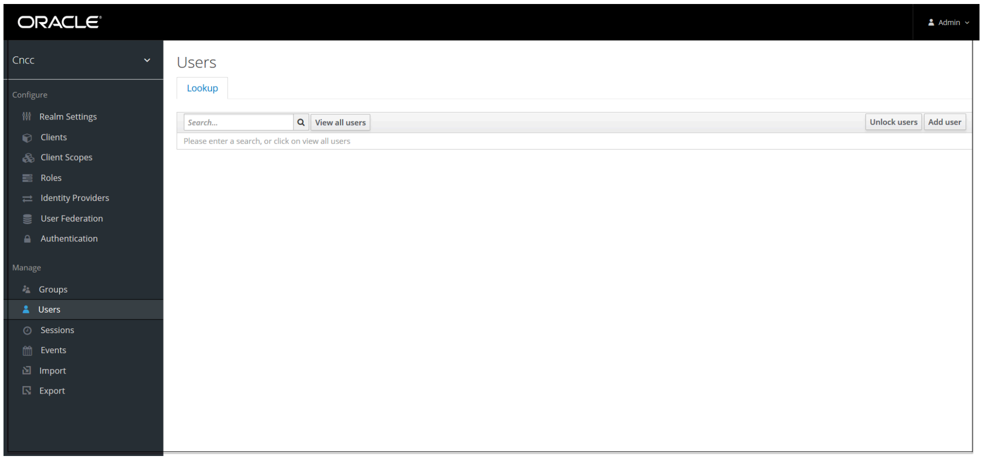
Frontend URL:

Enabled: ON

User-Managed Access: OFF

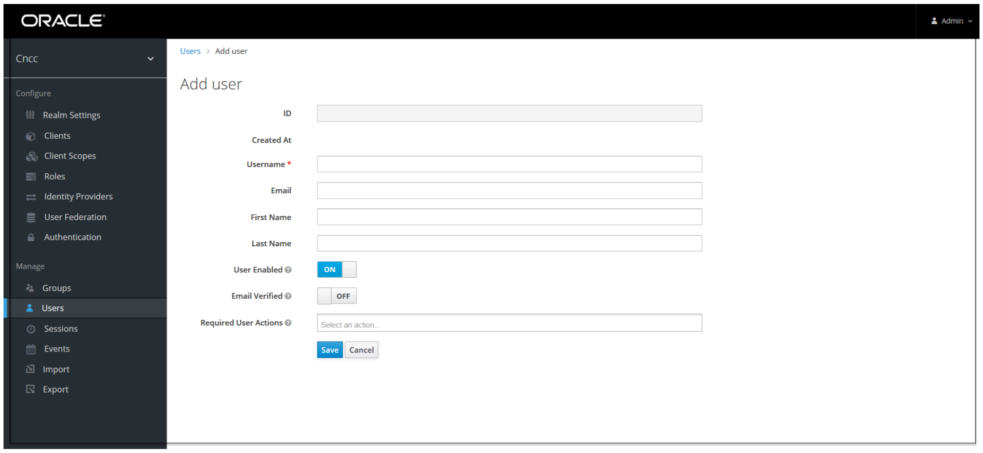
Endpoints: OpenID Endpoint Configuration
SAML 2.0 Identity Provider Metadata

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



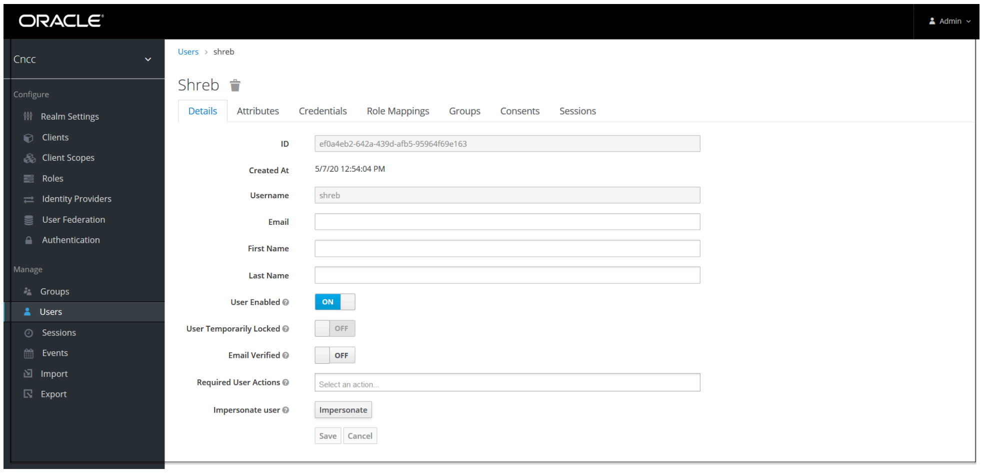
Please enter a search, or click on view all users

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



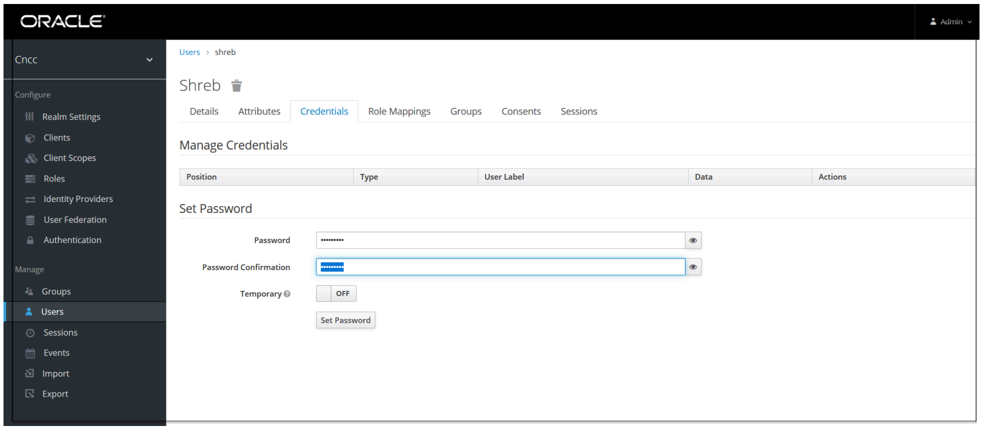
ID	
Created At	
Username *	
Email	
First Name	
Last Name	
User Enabled	ON
Email Verified	OFF
Required User Actions	Select an action...

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



ID	ef0a4eb2-642a-439d-afb5-95964f69e163
Created At	5/7/20 12:54:04 PM
Username	shreb
Email	
First Name	
Last Name	
User Enabled	ON
User Temporarily Locked	OFF
Email Verified	OFF
Required User Actions	Select an action...
Impersonate user	Impersonate

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.



Manage Credentials

Position	Type	User Label	Data	Actions

Set Password

Password: •••••••••

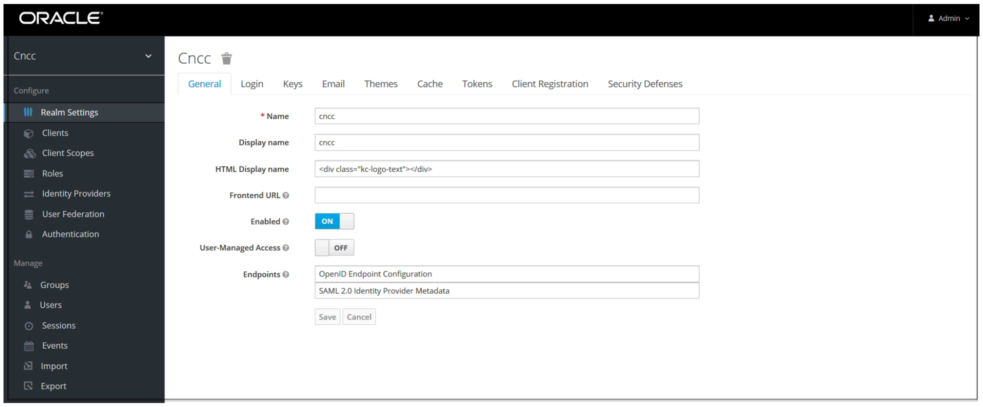
Password Confirmation: •••••••••

Temporary: OFF

> **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**.



* Name: cncc

Display name: cncc

HTML Display name: <div class="kc-logo-text"></div>

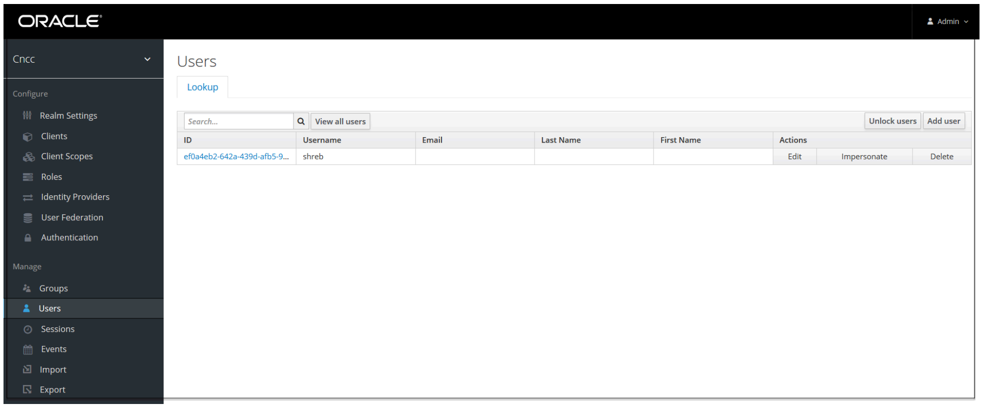
Frontend URL:

Enabled: ON

User-Managed Access: OFF

Endpoints: OpenID Endpoint Configuration
SAML 2.0 Identity Provider Metadata

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



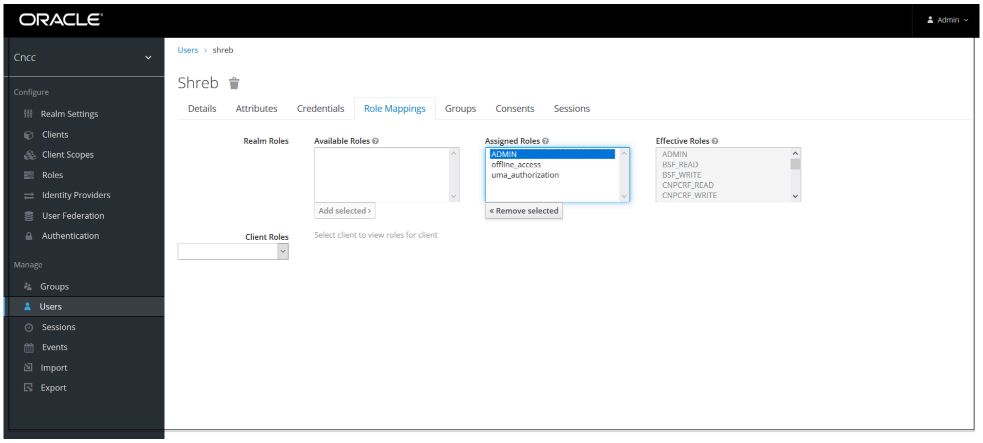
ID	Username	Email	Last Name	First Name	Actions		
ef0a4eb2-642a-439d-afb5-9...	shreb				Edit	Impersonate	Delete

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 SAML SSO with CNC Console IAM

Overview

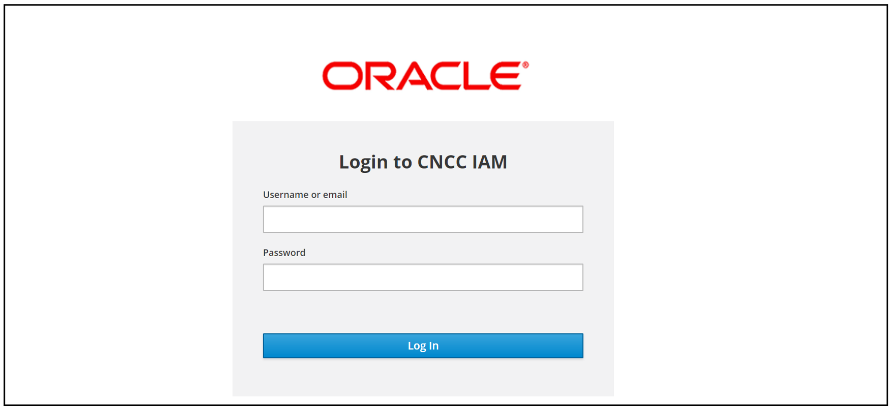
Security Assertion Markup Language (SAML) is an open standard that allows identity providers (IdP) to pass authorization credentials to service providers (SP). The identity provider authenticates the user and returns the assertion information about the authenticated user and the authentication event 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.

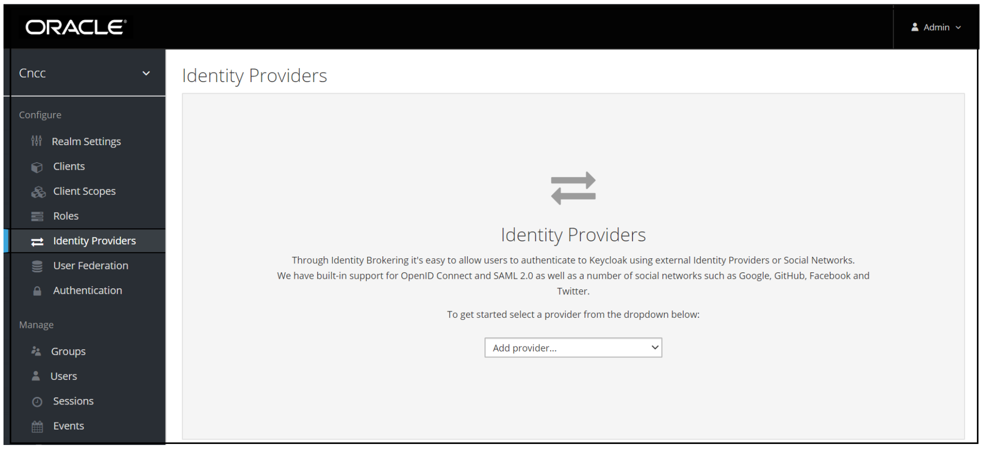
Configuring SAML Identity Provider in CNCC IAM

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://<cncc-iam-ingress-extrenal-ip>:<cncc-iam-ingress-service-
port>
Example: http://cncc-iam-ingress-
gateway.cncc.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.

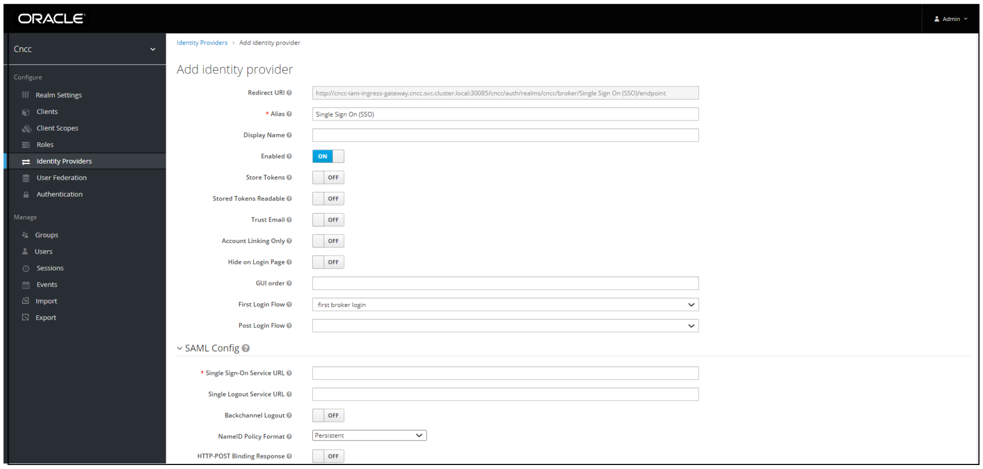


Identity Providers

Through Identity Brokering it's easy to allow users to authenticate to Keycloak using external Identity Providers or Social Networks. We have built-in support for OpenID Connect and SAML 2.0 as well as a number of social networks such as Google, GitHub, Facebook and Twitter.

To get started select a provider from the dropdown below:

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



Add identity provider

Redirect URI	http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/cncc/auth/realms/cncc/broker/Single Sign On (SSO)/endpoint
* Alias	Single Sign On (SSO)
Display Name	
Enabled	ON
Store Tokens	OFF
Stored Tokens Readable	OFF
Trust Email	OFF
Account Linking Only	OFF
Hide on Login Page	OFF
GUI order	
First Login Flow	first broker login
Post Login Flow	

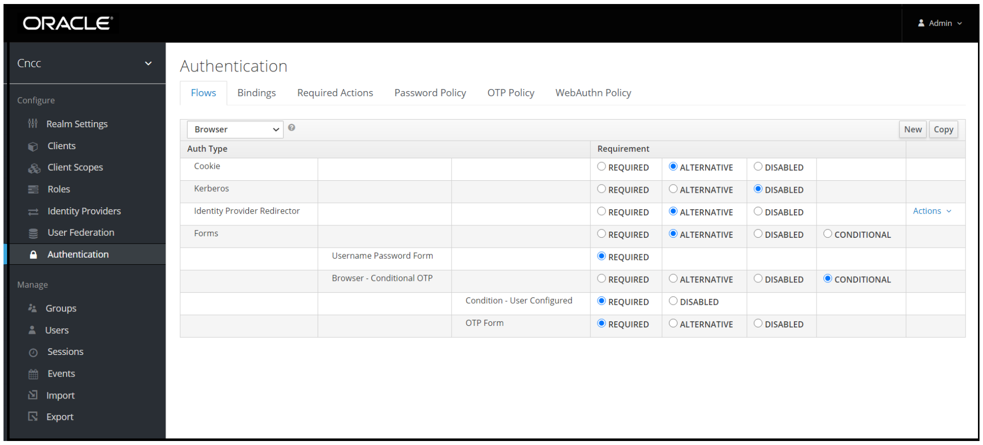
SAML Config

* Single Sign-On Service URL	
Single Logout Service URL	
Backchannel Logout	OFF
NameID Policy Format	Persistent
HTTP-POST Binding Response	OFF

> **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.

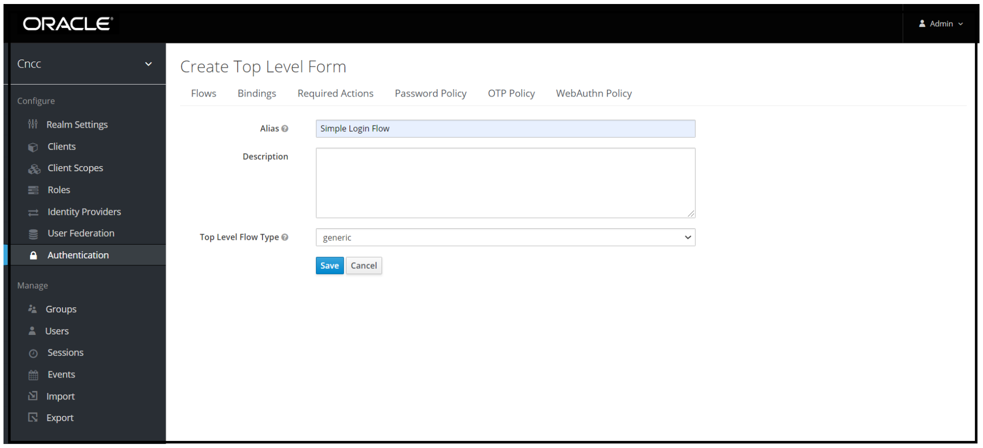
Click **Import** and **Save.**The other required fields will be filled in automatically.

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



Auth Type			Requirement				
Cookie			○ REQUIRED	● ALTERNATIVE	○ DISABLED		
Kerberos			○ REQUIRED	○ ALTERNATIVE	● DISABLED		
Identity Provider Redirector			○ REQUIRED	● ALTERNATIVE	○ DISABLED		Actions
Forms			○ REQUIRED	● ALTERNATIVE	○ DISABLED	○ CONDITIONAL	
	Username Password Form		● REQUIRED				
	Browser - Conditional OTP		○ REQUIRED	○ ALTERNATIVE	○ DISABLED	● CONDITIONAL	
		Condition - User Configured	● REQUIRED	○ DISABLED			
		OTP Form	● REQUIRED	○ ALTERNATIVE	○ DISABLED		

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



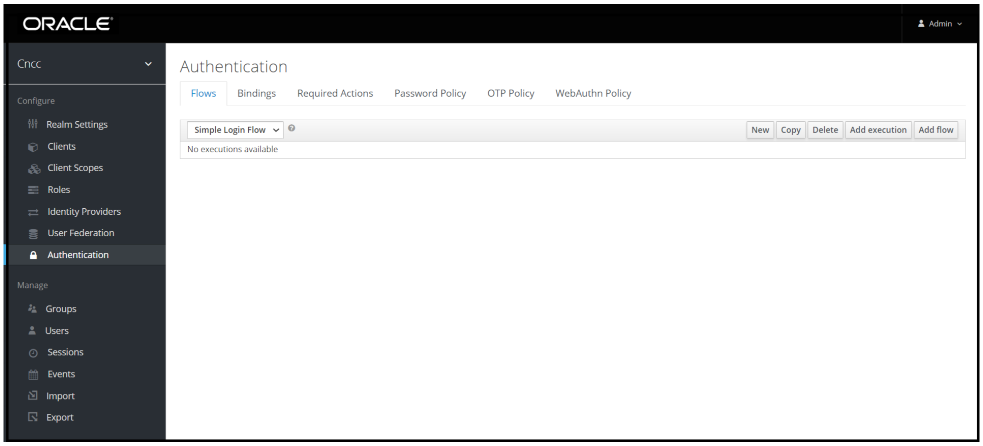
Alias: Simple Login Flow

Description:

Top Level Flow Type: generic

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

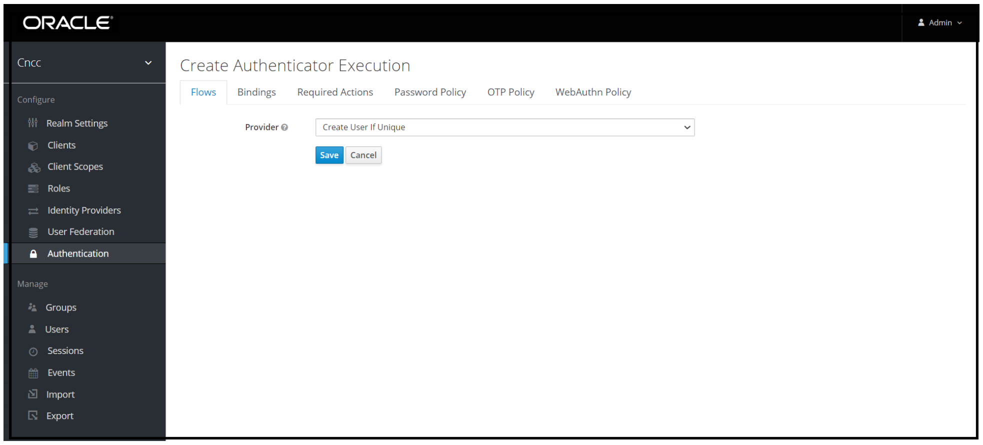
6. The **Authentication** screen with the newly created custom flow selected in the drop down list appears. Click **Add Execution** in the right pane .



Simple Login Flow

No executions available

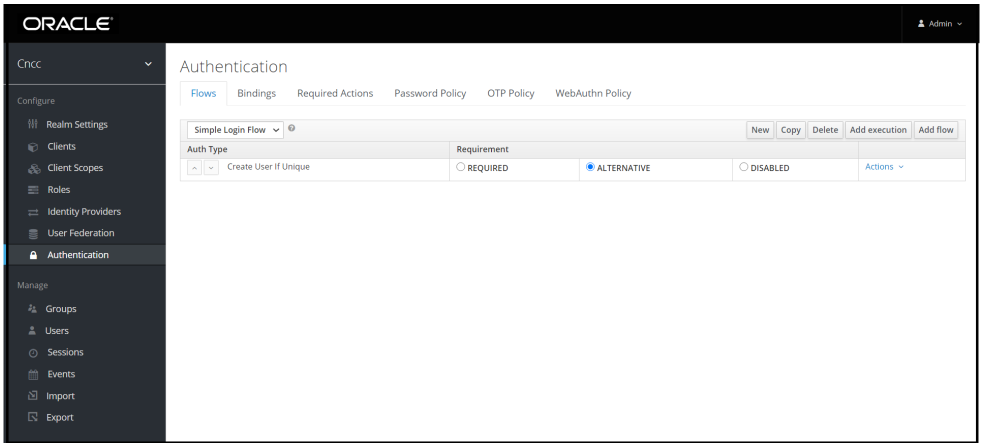
7. **Create Authenticator Execution** screen appears.



Provider: Create User If Unique

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

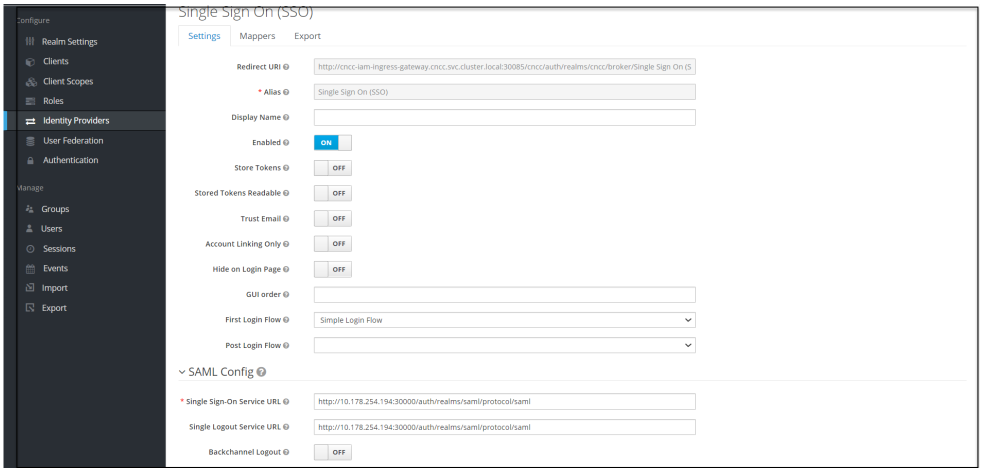
8. The **Authentication** screen with the newly created custom flow selected in the drop down. Under **Requirement** section, select **Alternative**.



Simple Login Flow

Auth Type	Requirement		
Create User If Unique	○ REQUIRED	● ALTERNATIVE	○ DISABLED

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



Single Sign On (SSO)

Redirect URI: http://cncc-iam-ingress-gateway.cncc.svc.cluster.local:30085/cncc/auth/realms/cncc/broker/Single Sign On (S

* Alias: Single Sign On (SSO)

Display Name:

Enabled: ON

Store Tokens: OFF

Stored Tokens Readable: OFF

Trust Email: OFF

Account Linking Only: OFF

Hide on Login Page: OFF

GUI order:

First Login Flow: Simple Login Flow

Post Login Flow:

SAML Config

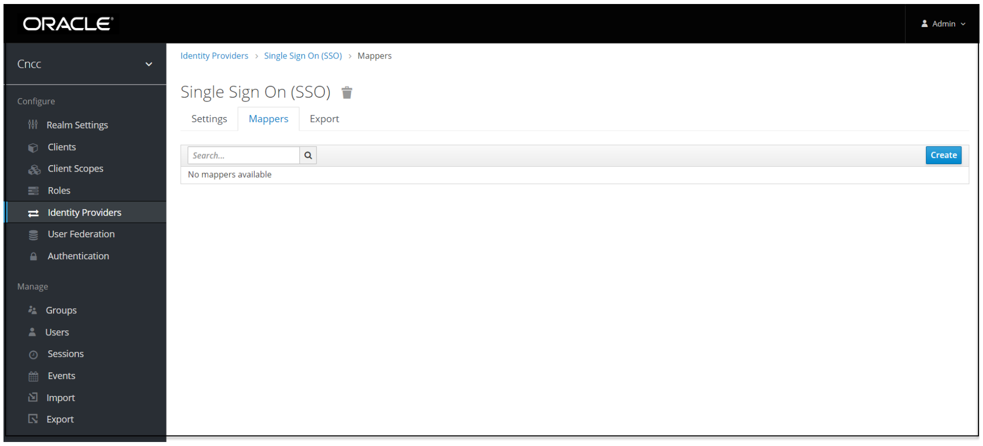
* Single Sign-On Service URL: http://10.178.254.194:30000/auth/realms/saml/protocol/saml

Single Logout Service URL: http://10.178.254.194:30000/auth/realms/saml/protocol/saml

Backchannel Logout: OFF

Mapping SAML IdP roles with CNCC IAM API roles

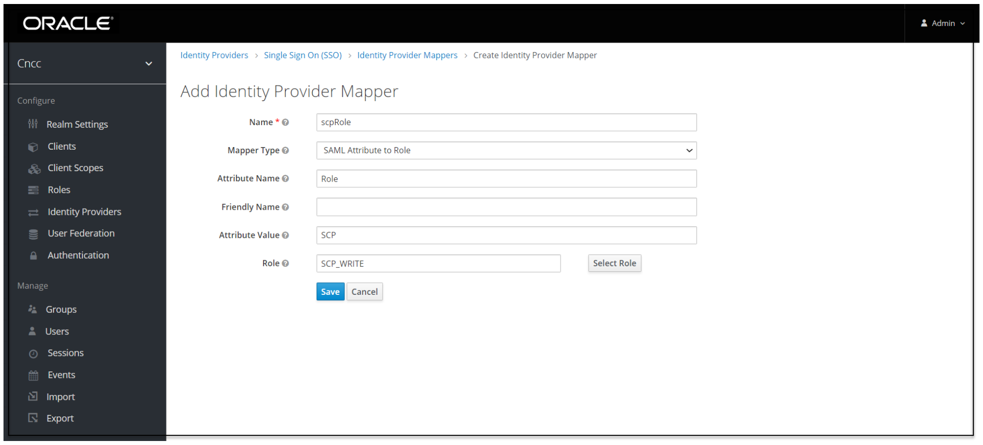
1. After saving SAML IdP configurations in CNCC IAM, select **Identity Providers** in the left pane and clock **Mappers** tab in the right pane. Click **Create.**



Single Sign On (SSO)

No mappers available

2. **Add Identity Provider Mapper** Screen appears.



Add Identity Provider Mapper

Name *: scpRole

Mapper Type: SAML Attribute to Role

Attribute Name: Role

Friendly Name:

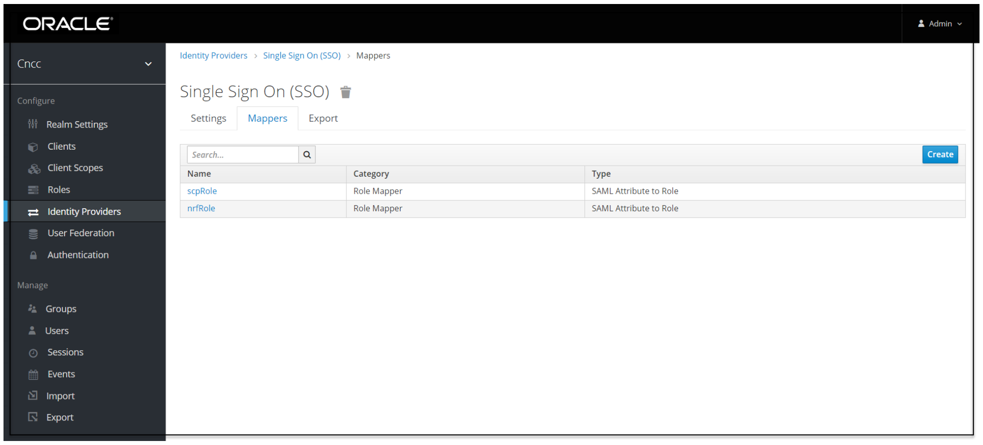
Attribute Value: SCP

Role: SCP_WRITE

- 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.

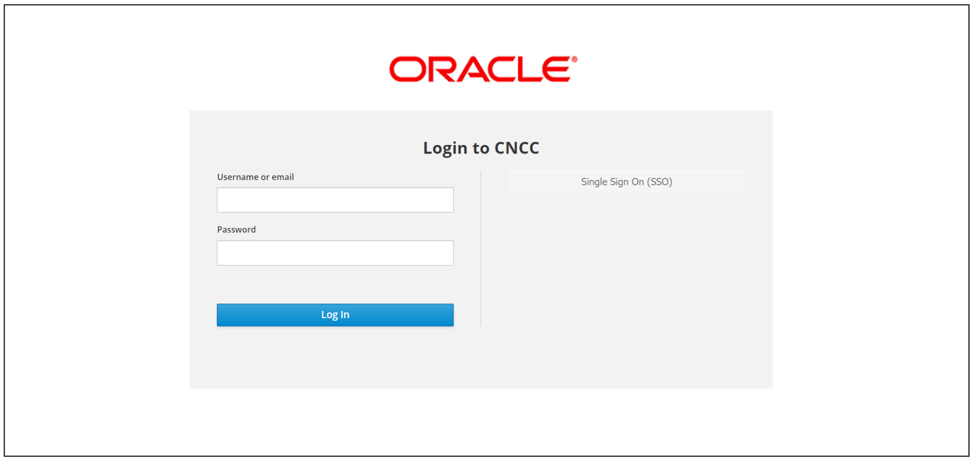


Name	Category	Type
scpRole	Role Mapper	SAML Attribute to Role
nrfRole	Role Mapper	SAML Attribute to Role

Accessing CNCC Core Application

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

```
http://<cncc-iam-ingress-extrenal-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.

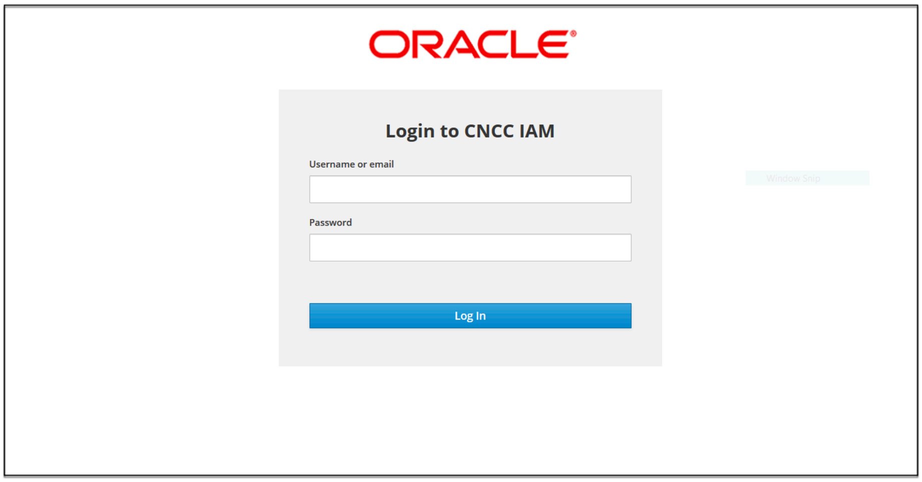
Sample LDAP ldif File

```
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
```

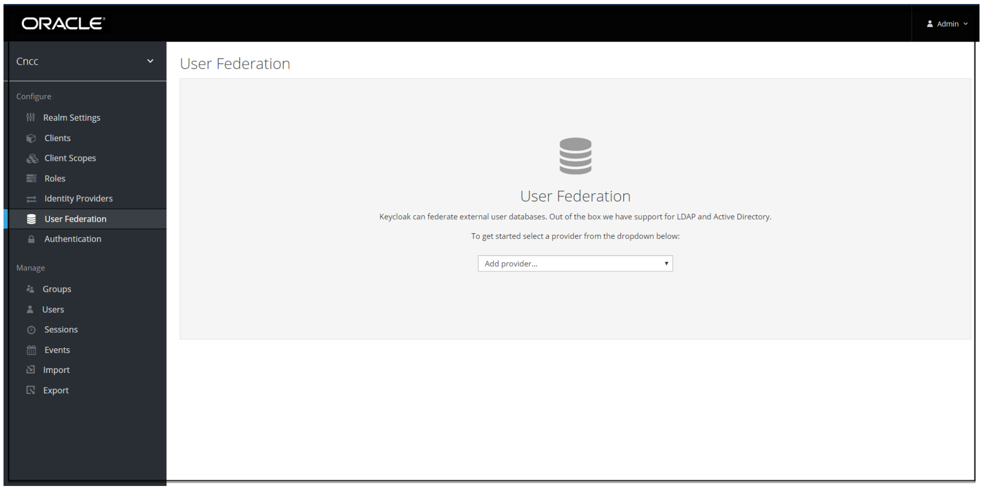
```
objectclass: groupOfUniqueNames
cn: admin
uniqueMember: uid=ben,ou=people,dc=oracle,dc=org
ou: admins

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.



User Federation

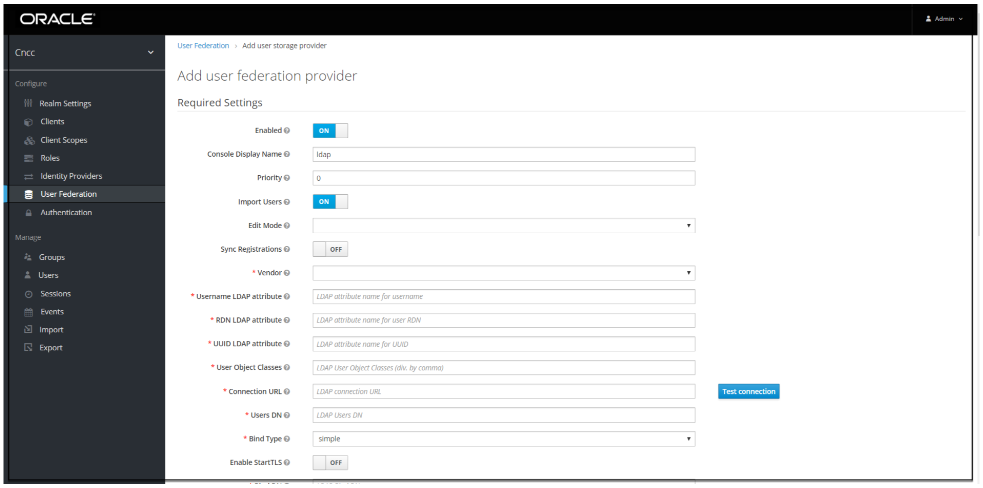
User Federation

Keycloak can federate external user databases. Out of the box we have support for LDAP and Active Directory.

To get started select a provider from the dropdown below:

Add provider... ▾

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



User Federation › Add user storage provider

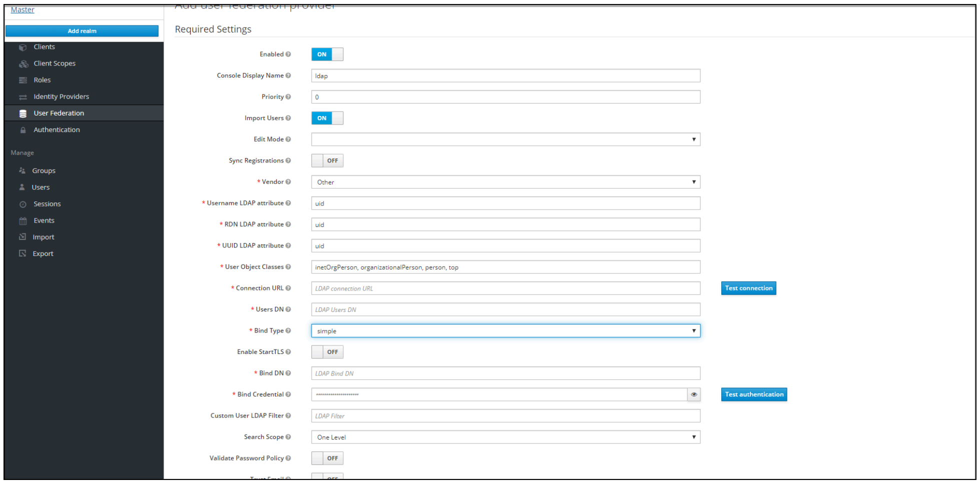
Add user federation provider

Required Settings

Enabled	ON
Console Display Name	ldap
Priority	0
Import Users	ON
Edit Mode	▾
Sync Registrations	OFF
* Vendor	▾
* Username LDAP attribute	LDAP attribute name for username
* RDN LDAP attribute	LDAP attribute name for user RDN
* UUID LDAP attribute	LDAP attribute name for UUID
* User Object Classes	LDAP User Object Classes (div. by comma)
* Connection URL	LDAP connection URL
* Users DN	LDAP Users DN
* Bind Type	simple ▾
Enable StartTLS	OFF

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 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.

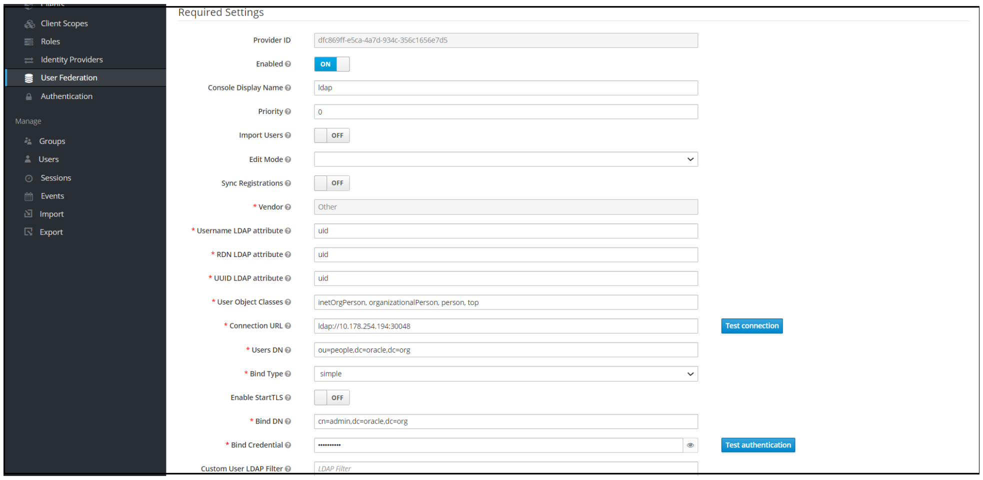


Required Settings

Enabled	ON
Console Display Name	ldap
Priority	0
Import Users	ON
Edit Mode	
Sync Registrations	OFF
* Vendor	Other
* Username LDAP attribute	uid
* RDN LDAP attribute	uid
* UUID LDAP attribute	uid
* User Object Classes	inetOrgPerson, organizationalPerson, person, top
* Connection URL	LDAP connection URL
* Users DN	LDAP Users DN
* Bind Type	simple
Enable StartTLS	OFF
* Bind DN	LDAP Bind DN
* Bind Credential	••••••••••••••••••••
Custom User LDAP Filter	LDAP Filter
Search Scope	One Level
Validate Password Policy	OFF

- 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".

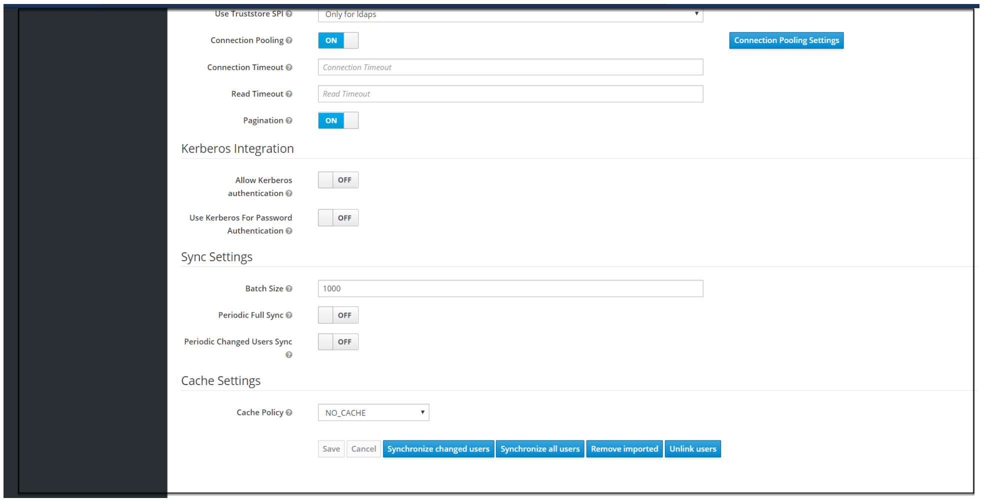
5. After filling the required fields, the screen appears as below. Click **Save**.



Required Settings

Provider ID	dfc869ff-e5ca-4a7d-934c-356c1656e7d5
Enabled	ON
Console Display Name	ldap
Priority	0
Import Users	OFF
Edit Mode	
Sync Registrations	OFF
* Vendor	Other
* Username LDAP attribute	uid
* RDN LDAP attribute	uid
* UUID 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	OFF
* Bind DN	cn=admin,dc=oracle,dc=org
* Bind Credential	••••••••••
Custom User LDAP Filter	LDAP Filter

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



Use Truststore SPI: Only for ldaps
Connection Pooling: ON
Connection Timeout: Connection Timeout
Read Timeout: Read Timeout
Pagination: ON

Kerberos Integration

Allow Kerberos authentication: OFF
Use Kerberos For Password Authentication: OFF

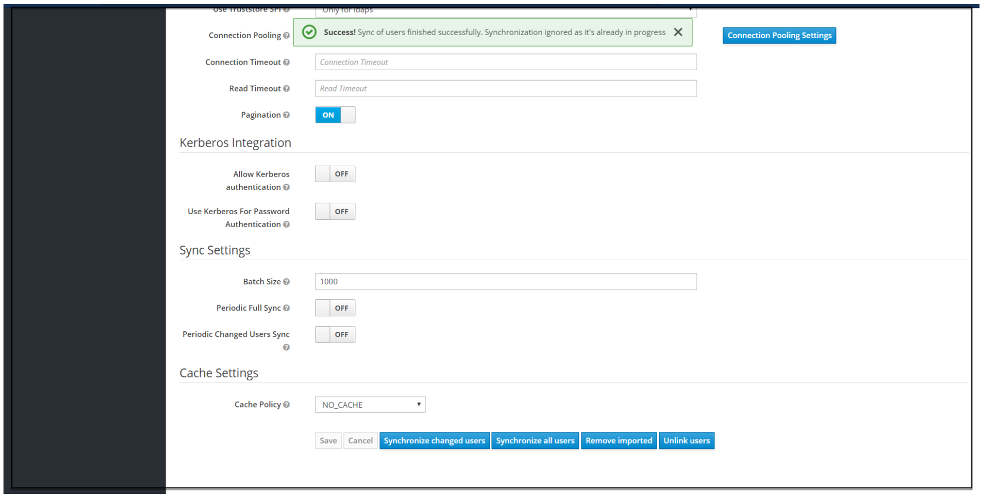
Sync Settings

Batch Size: 1000
Periodic Full Sync: OFF
Periodic Changed Users Sync: OFF

Cache Settings

Cache Policy: NO_CACHE

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.

   Success! Sync of users finished successfully. Synchronization ignored as it's already in progress

   Use Truststore SPI: Only for ldaps
   Connection Pooling
   Connection Timeout: Connection Timeout
   Read Timeout: Read Timeout
   Pagination: ON

   Kerberos Integration

   Allow Kerberos authentication: OFF
   Use Kerberos For Password Authentication: OFF

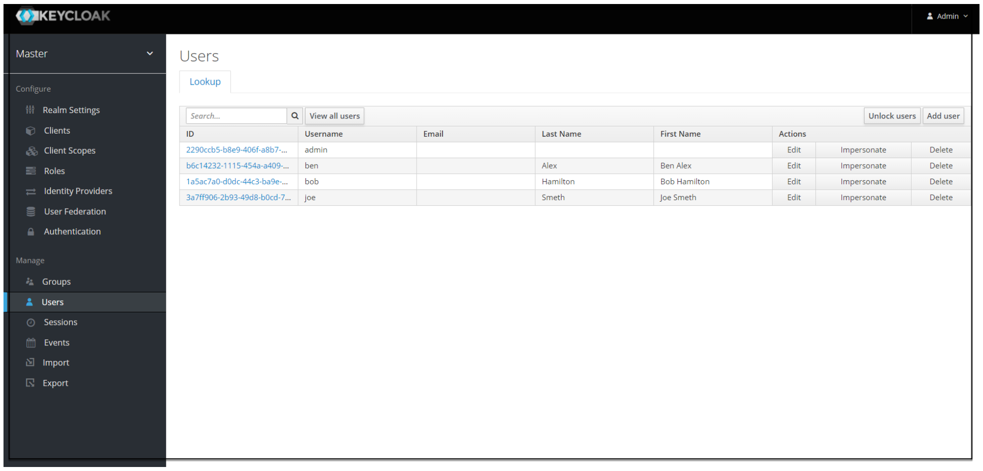
   Sync Settings

   Batch Size: 1000
   Periodic Full Sync: OFF
   Periodic Changed Users Sync: OFF

   Cache Settings

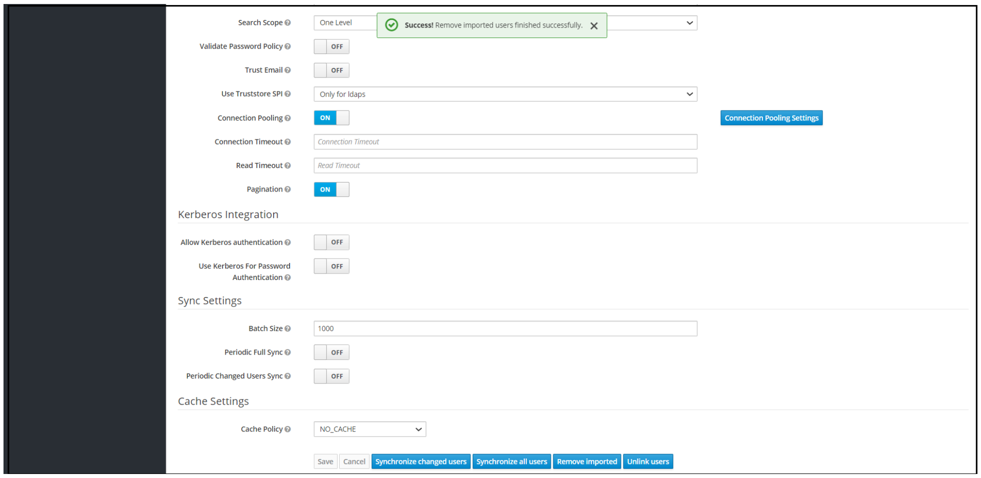
   Cache Policy: NO_CACHE

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.



ID	Username	Email	Last Name	First Name	Actions		
2290ccb5-b8e9-406f-a8b7-...	admin				Edit	Impersonate	Delete
b6c14232-1115-454a-a409-...	ben		Alex	Ben Alex	Edit	Impersonate	Delete
1a5ac7a0-d0dc-44c3-ba9e-...	bob		Hamilton	Bob Hamilton	Edit	Impersonate	Delete
3a7ff906-2b93-49d8-b0cd-7...	joe		Smeth	Joe Smeth	Edit	Impersonate	Delete

9. 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.



Success! Remove imported users finished successfully.

Search Scope: One Level
Validate Password Policy: OFF
Trust Email: OFF
Use Truststore SPI: Only for ldaps
Connection Pooling: ON
Connection Timeout: Connection Timeout
Read Timeout: Read Timeout
Pagination: ON

Kerberos Integration

Allow Kerberos authentication: OFF
Use Kerberos For Password Authentication: OFF

Sync Settings

Batch Size: 1000
Periodic Full Sync: OFF
Periodic Changed Users Sync: OFF

Cache Settings

Cache Policy: NO_CACHE

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

Group LDAP Mapper and Role Assignment

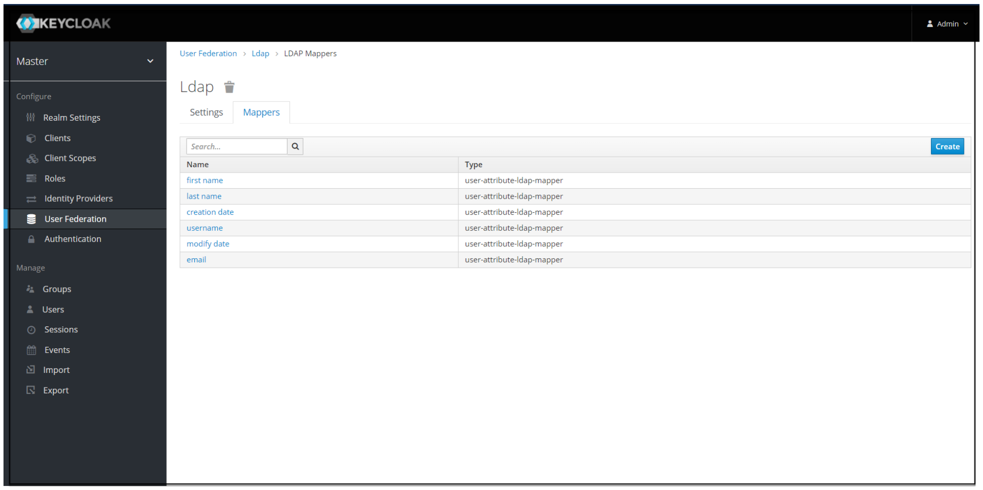
When an LDAP Federation provider is created, CNC Console-IAM provides a set of built-in mappers for this provider. User can change this set and create a new mapper or update/delete existing ones.

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.

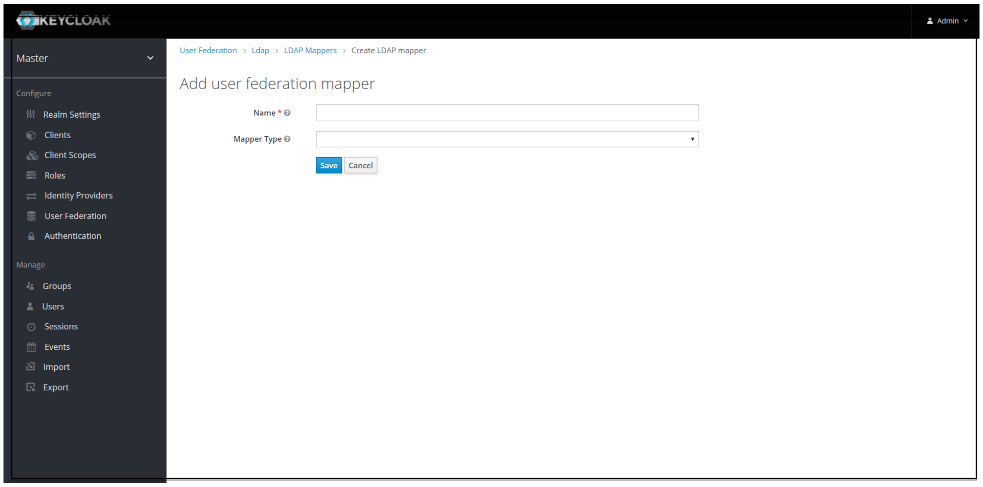
To add Group-Mapper and assign roles:

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

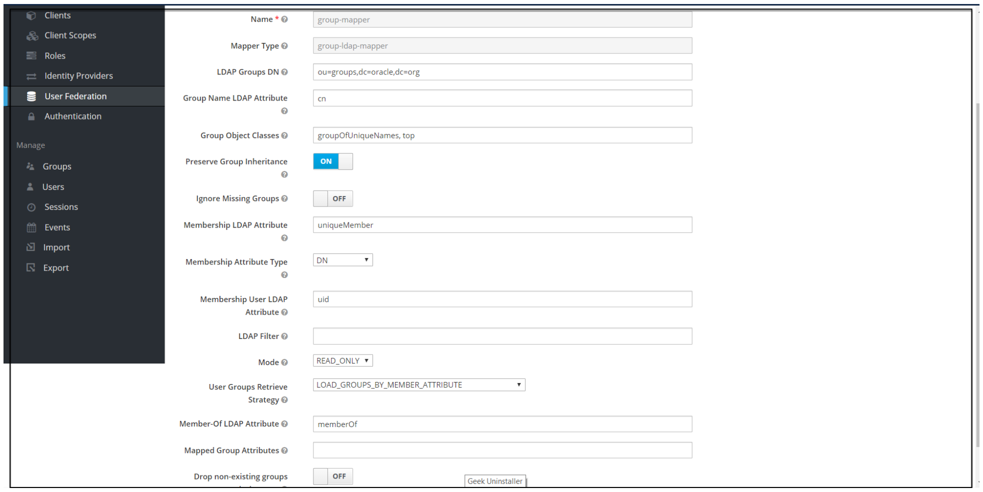


Name	Type
first name	user-attribute-ldap-mapper
last name	user-attribute-ldap-mapper
creation date	user-attribute-ldap-mapper
username	user-attribute-ldap-mapper
modify date	user-attribute-ldap-mapper
email	user-attribute-ldap-mapper

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.**



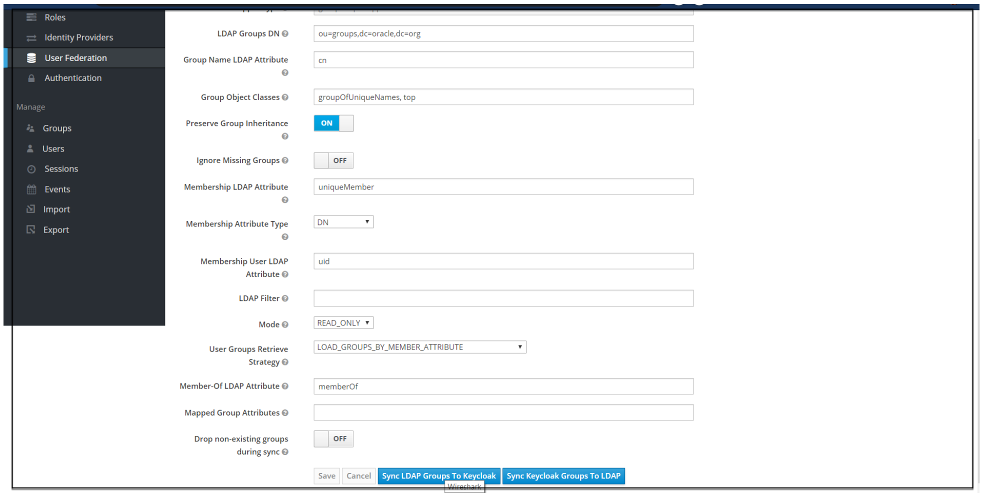
The following screen appears.



Name *	group-mapper
Mapper Type	group-ldap-mapper
LDAP Groups DN	ou=groups,dc=oracle,dc=org
Group Name LDAP Attribute	cn
Group Object Classes	groupOfUniqueNames, top
Preserve Group Inheritance	ON
Ignore Missing Groups	OFF
Membership LDAP Attribute	uniqueMember
Membership Attribute Type	DN
Membership User LDAP Attribute	uid
LDAP Filter	
Mode	READ_ONLY
User Groups Retrieve Strategy	LOAD_GROUPS_BY_MEMBER_ATTRIBUTE
Member-Of LDAP Attribute	memberOf
Mapped Group Attributes	
Drop non-existing groups	OFF

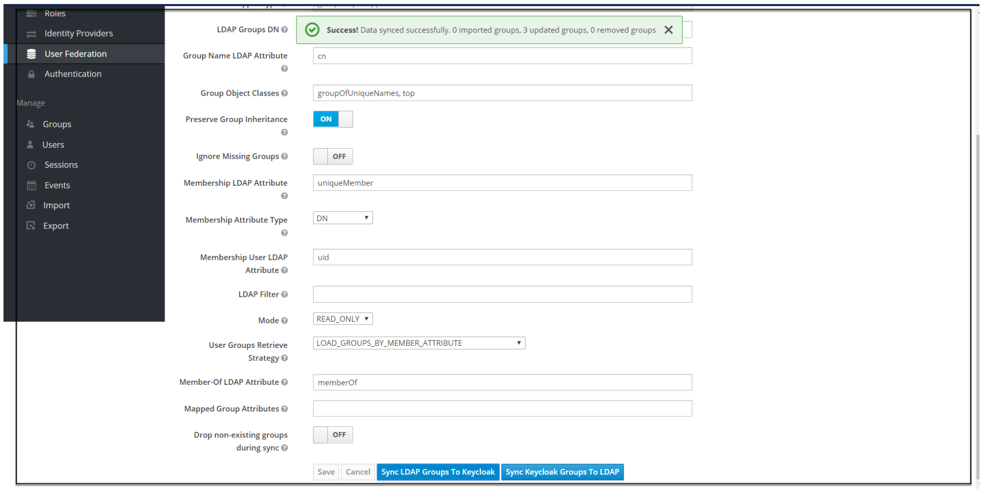
> **Note:**
>
> When selected, default values will be set by cncc-iam. But you need change some values based on your ldap records.

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.**



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	
Mode	READ_ONLY
User Groups Retrieve Strategy	LOAD_GROUPS_BY_MEMBER_ATTRIBUTE
Member-Of LDAP Attribute	memberOf
Mapped Group Attributes	
Drop non-existing groups during sync	OFF

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

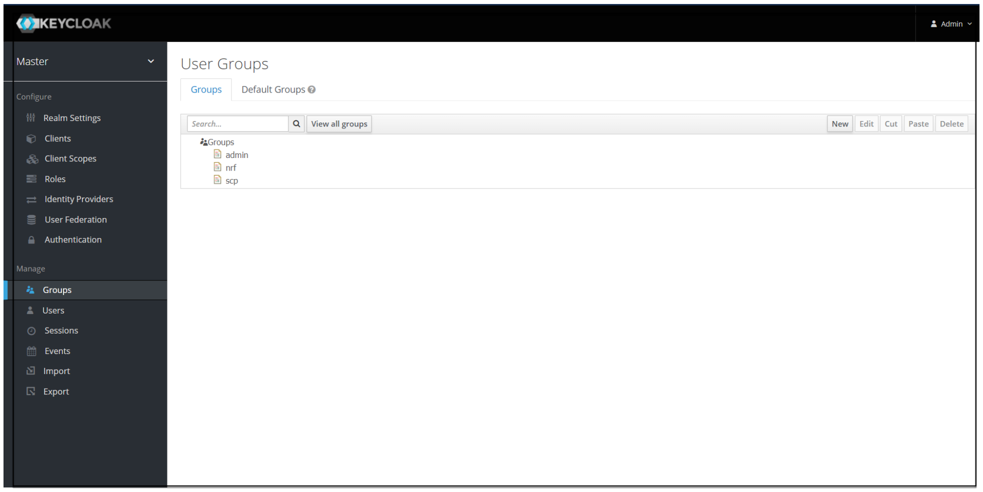


Success! Data synced successfully. 0 imported groups, 3 updated groups, 0 removed groups

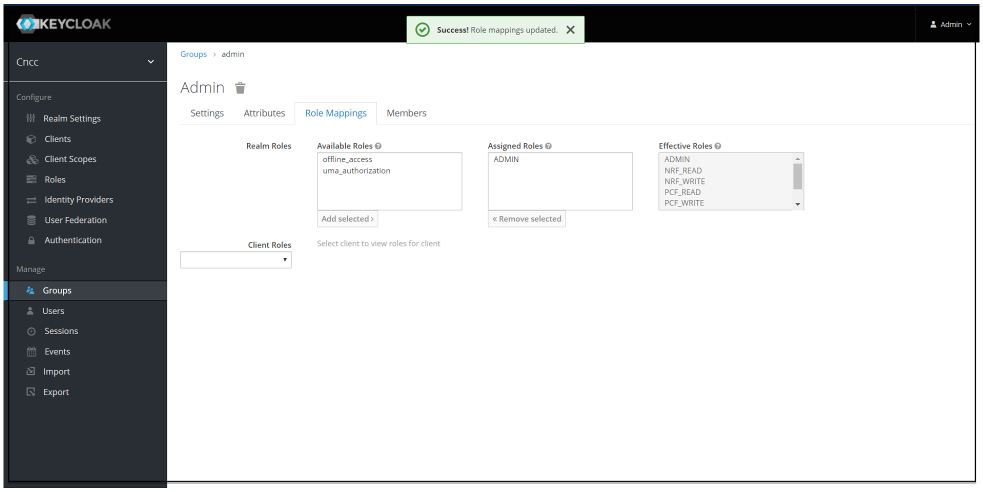
LDAP Groups DN	
Group Name LDAP Attribute	cn
Group Object Classes	groupOfUniqueNames, top
Preserve Group Inheritance	ON
Ignore Missing Groups	OFF
Membership LDAP Attribute	uniqueMember
Membership Attribute Type	DN
Membership User LDAP Attribute	uid
LDAP Filter	
Mode	READ_ONLY
User Groups Retrieve Strategy	LOAD_GROUPS_BY_MEMBER_ATTRIBUTE
Member-Of LDAP Attribute	memberOf
Mapped Group Attributes	
Drop non-existing groups during sync	OFF

> **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**.


Success! Role mappings updated.

- 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.

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-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":
   "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJHS1N4WVhoWlExRVh
   ```

```
rOVE5RTR3STN4WG9LcHI2RW5yOFJCdGlMVndPV0JZIn0.eyJqdGkiOiIwMTQzYzNhZC1
kNjE3LTQyNTYtYTA2My01NGYxNGI5MDQ5MWMiLCJleHAiOjE1ODQ3MDAyMzUsIm5iZiI
6MCwiaWF0IjoxNTg0Njk5OTM1LCJpc3MiOiJodHRwOi8vMTAuNzUuMjEzLjYwOjMwMDI
0L2NuY2MvYXV0aC9yZWFsbXMvY25jYyIsImF1ZCI6ImFjY291bnQiLCJzdWIiOiIwMTJ
lNDI2OS02YzYwLTQzNWItYWExNC0yYWI3NmJjOWI1MjMiLCJ0eXAiOiJCZWFyZXIiLCJ
henAiOiJhcGktZ2F0ZXdheSIsImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6IjZ
jNDJkOTc4LTE0YWMtNDc5My1hMWUzLTc4OWNmYmRiMmI3NCIsImFjciI6IjEiLCJyZWF
sbV9hY2Nlc3MiOnsicm9sZXMiOlsiUENGX1JFQUQiLCJQQ0ZfV1JJVEUiLCJTQ1BfUkV
BRCIsIm9mZmxpbmVfYWNjZXNzIiwiTlJGX1dSSVRFIiwiQURNSU4iLCJ1bWFfYXV0aG9
yaXphdGlvbiIsIk5SRl9SRUFEIiwiU0NQX1dSSVRFIl19LCJyZXNvdXJjZV9hY2Nlc3M
iOnsiYWNjb3VudCI6eyJyb2xlcyI6WyJtYW5hZ2UtYWNjb3VudCIsIm1hbmFnZS1hY2N
vdW50LWxpbmtzIiwidmlldy1wcm9maWxlIl19fSwic2NvcGUiOiJlbWFpbCBwcm9maWx
lIiwiZW1haWxfdmVyaWZpZWQiOmZhbHNlLCJuYW1lIjoiU2hyZXlhcyBCIiwicHJlZmV
ycmVkX3VzZXJuYW1lIjoic2hyZWIiLCJnaXZlbl9uYW1lIjoiU2hyZXlhcyIsImZhbWl
seV9uYW1lIjoiQiIsImVtYWlsIjoic2hyZXlhcy5iQG9yYWNsZS5jb20ifQ.fXYyjmAb
SSIFlLr2ZBEX2pfKrE_vr6Zbj8ta-
l_tKlv2gTX1J3ehScg_m30swpWU7UojuFkyc8CfNZL2Z9mcs7zbq_zA7ZTlaWA_Agmeo
XWapicX2wALT_YDU6Z3H7L9x1C1Ulp8aTBIBHPv2J-
zgkrFDtk83NeKunKEGlEZpp-9MGDLQ5a8QX6SAUo-
Fe6hNgF1vP0d7LCyjWvu6UvoeG_Fuxsi4xEVHcbSen8M3eueAt7xN7akhXZ_4PgWnxsW
vQVqtTzsY6O-
WyUjUiwtaTvpX0dPVVeeNDvWMY_0q0KvF_nnE3_wQtE8bu_LcCZYwDQJJTloj2PJ8y1W
jO9l2Q",
    "expires_in": 300,
    "refresh_expires_in": 1800,
    "refresh_token":
"eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICI3YTFlYjcyZi00MWE
1LTRkMTEtYjRmZS01NDZjZGU5NjY2MTUifQ.eyJqdGkiOiJmYjAwZTY2OC0xZTkyLTRl
MTUtYTVlMS1jZjgxNDFkMjllNDMiLCJleHAiOjE1ODQ3MDE3MzUsIm5iZiI6MCwiaWF0
IjoxNTg0Njk5OTM1LCJpc3MiOiJodHRwOi8vMTAuNzUuMjEzLjYwOjMwMDI0L2NuY2Mv
YXV0aC9yZWFsbXMvY25jYyIsImF1ZCI6Imh0dHA6Ly8xMC43NS4yMTMuNjA6MzAwMjQv
Y25jYy9hdXRoL3JlYWxtcy9jbmNjIiwic3ViIjoiMDEyZTQyNjktNmM2MC00MzViLWFh
MTQtMmFiNzZiYzliNTIzIiwidHlwIjoiUmVmcmVzaCIsImF6cCI6ImFwaS1nYXRld2F5
IiwiYXV0aF90aW1lIjowLCJzZXNzaW9uX3N0YXRlIjoiNmM0MmQ5NzgtMTRhYy00Nzkz
LWExZTMtNzg5Y2ZiZGIyYjc0IiwicmVhbG1fYWNjZXNzIjp7InJvbGVzIjpbIlBDRl9S
RUFEIiwiUENGX1dSSVRFIiwiU0NQX1JFQUQiLCJvZmZsaW5lX2FjY2VzcyIsIk5SRl9X
UklURSIsIkFETUlOIiwidW1hX2F1dGhvcml6YXRpb24iLCJOUkZfUkVBRCIsIlNDUF9X
UklURSJdfSwicmVzb3VyY2VfYWNjZXNzIjp7ImFjY291bnQiOnsicm9sZXMiOlsibWFu
YWdlLWFjY291bnQiLCJtYW5hZ2UtYWNjb3VudC1saW5rcyIsInZpZXctcHJvZmlsZSJd
fX0sInNjb3BlIjoiZW1haWwgcHJvZmlsZSJ9.l8w3j1gMNgblKSYdvCmJQfg6yIfkdKn
mFb5vKPF-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':
   'eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICI3YTFlYjcyZi00MWE
   1LTRkMTEtYjRmZS01NDZjZGU5NjY2MTUifQ.eyJqdGkiOiJmYjAwZTY2OC0xZTkyLTRl
   MTUtYTVlMS1jZjgxNDFkMjllNDMiLCJleHAiOjE1ODQ3MDE3MzUsIm5iZiI6MCwiaWF0
   IjoxNTg0Njk5OTM1LCJpc3MiOiJodHRwOi8vMTAuNzUuMjEzLjYwOjMwMDI0L2NuY2Mv
   YXV0aC9yZWFsbXMvY25jYyIsImF1ZCI6Imh0dHA6Ly8xMC43NS4yMTMuNjA6MzAwMjQv
   Y25jYy9hdXRoL3JlYWxtcy9jbmNjIiwic3ViIjoiMDEyZTQyNjktNmM2MC00MzViLWFh
   MTQtMmFiNzZiYzliNTIzIiwidHlwIjoiUmVmcmVzaCIsImF6cCI6ImFwaS1nYXRld2F5
   IiwiYXV0aF90aW1lIjowLCJzZXNzaW9uX3N0YXRlIjoiNmM0MmQ5NzgtMTRhYy00Nzkz
   LWExZTMtNzg5Y2ZiZGIyYjc0IiwicmVhbG1fYWNjZXNzIjp7InJvbGVzIjpbIlBDRl9S
   RUFEIiwiUENGX1dSSVRFIiwiU0NQX1JFQUQiLCJvZmZsaW5lX2FjY2VzcyIsIk5SRl9X
   UklURSIsIkFETUlOIiwidW1hX2F1dGhvcml6YXRpb24iLCJOUkZfUkVBRCIsIlNDUF9X
   UklURSJdfSwicmVzb3VyY2VfYWNjZXNzIjp7ImFjY291bnQiOnsicm9sZXMiOlsibWFu
   YWdlLWFjY291bnQiLCJtYW5hZ2UtYWNjb3VudC1saW5rcyIsInZpZXctcHJvZmlsZSJd
   fX0sInNjb3BlIjoiZW1haWwgcHJvZmlsZSJ9.l8w3j1gMNgblKSYdvCmJQfg6yIfkdKn
   mFb5vKPF-ZIg',
   'grant_type': 'refresh_token'
   ```

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

6
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 filelds. 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 6-1   Log Details

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

Table 6-1 (Cont.) Log Details

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

CNCC Message Format

Table 6-2 CNCC Message Format

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

Table 6-2 (Cont.) CNCC Message Format

Name	Description	Example	Possible Values
resourceType	Indicates what is the resource being requested for	resourceType=SCP	CM_SERVICE (For default route) CNCC (For User Login Activity) SCP UDR NRF PCF ... all CNCC supported NF's
userId	Id of the user. Basically to know who triggered request/action	userId=3314f54f-08bf-489d-b395-27bf56da1262	
username	Name of the user	username=shreb	
status	Http status of the response.	status=200 OK	
operationType	HTTP method of the request	operationType=GET	
scheme	Indicates the scheme of the request	scheme=http	
remoteAddress	Remote Address associated with request. i.e remote address where this request is connected to, if available.	remoteAddress=/192.168.219.64:53587	
localAddress	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	
resourcePath	Request uri	resourcePath=/soothsayer/v1/canaryrelease/	
queryParams	Query parameters associated with request	queryParams={form_id=9, page=1, view_id=78}	

Table 6-2 (Cont.) CNCC Message Format

Name	Description	Example	Possible Values
headers	Headers associated with request or response	headers={Accept=*/*, X-Requested-With=XMLHttpRequest, User-Agent=Mozilla/5.0 (Windows NT 10.0; WOW64; rv:68.0) Gecko/20100101 Firefox/68.0, Connection=keep-alive, Host=cncc-core-ingress-gateway.cncc.svc.cluster.local:30075, Accept-Language=en-US,en;q=0.5, Accept-Encoding=gzip, deflate, DNT=1, Content-Type=application/json; charset=utf-8}	
payload	Payload/Data associated with request or response	payload=[{"serviceName":"n5g-eir-eic","canaryReleaseFlag":true,"apiFullVersion":"2.0.0","canaryTraffic":5}	
authenticationType	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-eecf-406b-
a7eb-92e249c2beeb,
                              username=admin, 
                              authSessionParentId=5aca4960-
eecf-406b-a7eb-92e249c2beeb,
                              authSessionTabId=8sruELA1WWs
```

- Login with correct credential

```
04:57:24,581 INFO  [org.keycloak.events] (default task-22)
                              type=LOGIN, realmId=master,
clientId=security-admin-console,
                              userId=d7cde46f-15e1-4ff8-a2cb-
c5825e481438, ipAddress=192.168.219.64,
                              auth_method=openid-connect,
auth_type=code, redirect_uri=http://10.75.225.28:31373/cncc/auth/admin/master/
console/, consent=no_consent_required,
                              code_id=5aca4960-eecf-406b-
a7eb-92e249c2beeb, username=admin,
                              authSessionParentId=5aca4960-
eecf-406b-a7eb-92e249c2beeb,
                              authSessionTabId=8sruELA1WWs
```

- User created

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

- Deleted user

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

- Admin Role removed for a user

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

- Admin Role added for a user

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

- Realm setting update

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

- Logout all session on keycloak

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

Examples of Security Logs

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

CNCC Core

- SCP request

```
logType=SECURITY, type=REQUEST, resourceType=SCP,
                                    userId=3314f54f-08bf-489d-
b395-27bf56da1262, username=shreb,
                                    operationType=GET, scheme=http,
                                    remoteAddress=/
192.168.219.64:53587,
                                    localAddress=cncc-core-ingress-
gateway.cncc.svc.cluster.local/<unresolved>:30075,
                                    resourcePath=/soothsayer/v1/
canaryrelease/, queryParams={},
                                    headers={Accept=*/*, X-
Requested-With=XMLHttpRequest,
                                    User-Agent=Mozilla/5.0 (Windows
NT 10.0; WOW64; rv:68.0)
                                    Gecko/20100101 Firefox/68.0,
Connection=keep-alive,
                                    Host=cncc-core-ingress-
gateway.cncc.svc.cluster.local:30075,
                                    Accept-Language=en-US,en;q=0.5,
Accept-Encoding=gzip, deflate,
                                    DNT=1, Content-Type=application/
json; charset=utf-8},
                                    payload={},
authenticationType=OAUTH
```

- SCP response

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

CNCC IAM

- Request

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

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

- Response

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

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

Accessing logs

This section gives information about how to access the logs.

The CNCC application logs can be accessed in following ways:

1. 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:

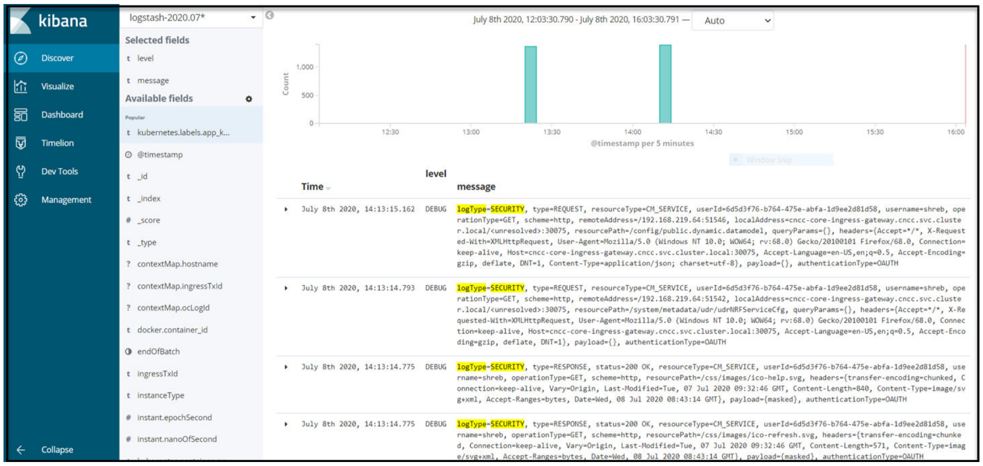
```
$ kubectl logs -f -n cncc cncc-core-ingress-gateway-77df795fb5-wv2sb
                              -c ingress-gateway (Security & Audit
Log)
```

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.



```
Time            level  message
July 8th 2020, 14:13:15.162  DEBUG  logType=SECURITY, type=REQUEST, resourceType=CM_SERVICE, userId=6d5d3f76-b764-475e-abfa-1d9ee2d81d58, username=shreb, ope
rationType=GET, scheme=http, remoteAddress=/192.168.219.64:51546, localAddress=cncc-core-ingress-gateway.cncc.svc.cluste
r.local/<unresolved>:30075, resourcePath=/config/public.dynamic.datamodel, queryParams={}, headers={Accept=*/*, X-Request
ed-With=XMLHttpRequest, User-Agent=Mozilla/5.0 (Windows NT 10.0; WOW64; rv:68.0) Gecko/20100101 Firefox/68.0, Connection=
keep-alive, Host=cncc-core-ingress-gateway.cncc.svc.cluster.local:30075, Accept-Language=en-US,en;q=0.5, Accept-Encoding=
gzip, deflate, DNT=1, Content-Type=application/json; charset=utf-8}, payload={}, authenticationType=OAUTH
July 8th 2020, 14:13:14.793  DEBUG  logType=SECURITY, type=REQUEST, resourceType=CM_SERVICE, userId=6d5d3f76-b764-475e-abfa-1d9ee2d81d58, username=shreb, ope
rationType=GET, scheme=http, remoteAddress=/192.168.219.64:51542, localAddress=cncc-core-ingress-gateway.cncc.svc.cluste
r.local/<unresolved>:30075, resourcePath=/system/metadata/udr/udrNRFServiceCfg, queryParams={}, headers={Accept=*/*, X-Re
quested-With=XMLHttpRequest, User-Agent=Mozilla/5.0 (Windows NT 10.0; WOW64; rv:68.0) Gecko/20100101 Firefox/68.0, Connec
tion=keep-alive, Host=cncc-core-ingress-gateway.cncc.svc.cluster.local:30075, Accept-Language=en-US,en;q=0.5, Accept-Enco
ding=gzip, deflate, DNT=1}, payload={}, authenticationType=OAUTH
July 8th 2020, 14:13:14.775  DEBUG  logType=SECURITY, type=RESPONSE, status=200 OK, resourceType=CM_SERVICE, userId=6d5d3f76-b764-475e-abfa-1d9ee2d81d58, use
rname=shreb, operationType=GET, scheme=http, resourcePath=/css/images/ico-help.svg, headers={transfer-encoding=chunked, C
onnection=keep-alive, Vary=Origin, Last-Modified=Tue, 07 Jul 2020 09:32:46 GMT, Content-Length=840, Content-Type=image/sv
g+xml, Accept-Ranges=bytes, Date=Wed, 08 Jul 2020 08:43:14 GMT}, payload={masked}, authenticationType=OAUTH
July 8th 2020, 14:13:14.775  DEBUG  logType=SECURITY, type=RESPONSE, status=200 OK, resourceType=CM_SERVICE, userId=6d5d3f76-b764-475e-abfa-1d9ee2d81d58, use
rname=shreb, operationType=GET, scheme=http, resourcePath=/css/images/ico-refresh.svg, headers={transfer-encoding=chunke
d, Connection=keep-alive, Vary=Origin, Last-Modified=Tue, 07 Jul 2020 09:32:46 GMT, Content-Length=571, Content-Type=imag
e/svg+xml, Accept-Ranges=bytes, Date=Wed, 08 Jul 2020 08:43:14 GMT}, payload={masked}, authenticationType=OAUTH
```

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.