

Oracle® Communications

Cloud Native Configuration Console REST Specifications Guide



Release 25.1.200
G35643-02
November 2025

ORACLE®

Copyright © 2023, 2025, Oracle and/or its affiliates.

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

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

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

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

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

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

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

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

Contents

1 Introduction

1.1	Overview	1
1.2	Reference	1

2 REST Specifications for Non OCI Deployment

2.1	Generate access token	1
2.2	Create a new user	2
2.3	Get users	3
2.4	Retrieve Specific User Information by Username	5
2.5	Get single user	6
2.6	Update Existing User	7
2.7	Delete the User	9
2.8	Set up a New Password for the User	10
2.9	Reset the Temporary Password for a User	11
2.10	Get Realm Level Roles	12
2.11	Get Realm-level Role Mappings	13
2.12	Add Realm-level Role Mappings to the User	15
2.13	Get Realm-level Role Mappings for a specific user	17
2.14	Remove All User Sessions Associated with the User	18
2.15	Get the Client details	19
2.16	Update the Client	20

3 REST Specification for OCI Deployment

3.1	Generate Access Token	1
3.2	Create a New User	1
3.3	Get All User	3
3.4	Get Specific User	5
3.5	Delete User	8
3.6	Set up a New Password for the User	9
3.7	Get all groups of the domain	11
3.8	Get all groups of a specific user	14
3.9	Create a group	19

3.10	Delete a group	23
3.11	Update group details	24
3.12	Assign group to a user	26
3.13	Assign multiple groups to a user	28

Preface

- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Conventions](#)

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Acronyms

The following table lists the acronyms and the terminologies used in the document:

Table Acronyms

Acronym	Description
A-CNCC Core	Agent CNC Console is a CNCC Core instance which manages local NF(s) and local OCCNE common services(s). A-CNCC is managed by M-CNCC. A-CNCC contains A-CNCC Core Ingress Gateway. A-CNCC has no IAM component. A-CNCC is also known as A-CNCC Core or aCncc Core.
AD	Active Directory
ASM	Aspen Service Mesh
BSF	Oracle Communications Cloud Native Core, Binding Support Function
CNC Console	Oracle Communications Cloud Native Configuration Console
cnDBTier	Oracle Communications Cloud Native Core, cnDBTier
CNE	Oracle Communications Cloud Native Core, Cloud Native Environment
CNI	Container Network Interface
CNLB	Cloud Native Load Balancer
CRD	Custom Resource Definitions
CRUD Operations	CREATE, READ, UPDATE, DELETE
CS	Common Service
ECDSA	Elliptic Curve Digital Signature Algorithm
EIR	Equipment Identity Register
GRR	Geo Replication Recovery
HTTPS	Hypertext Transfer Protocol Secure
IAM	Identity Access Management
Instance	NF or CNE common service managed by either M-CNCC Core or A-CNCC Core.
KPI	Key Performance Indicator
LCM	Lifecycle Management
LDAP	Lightweight Directory Access Protocol
LDAPS	Lightweight Directory Access Protocol (Over SSL)
MC	Multi Cluster. In multi cluster, a single CNCC can manage NF instances that access different Kubernetes clusters.
MCMI	Multiple Clusters Multiple Instances
M-CNCC	Manager CNC Console or mCncc is a CNC Console instance which manages multiple A-CNCC and local instances. Non OCI: M-CNCC has two components M-CNCC IAM and M-CNCC Core OCI: M-CNCC has only M-CNCC Core component. M-CNCC IAM is substituted with OCI IAM.

Table (Cont.) Acronyms

Acronym	Description
M-CNCC Core	Manager CNC Console Core or M-CNCC Core (also known as mCncc Core) is a core component of M-CNCC that provides GUI and API access portal for accessing NF and OCCNE common services. M-CNCC Core contains M-CNCC Core Ingress Gateway and M-CNCC Core back-end microservices.
M-CNCC IAM	Manager CNC Console IAM or M-CNCC IAM (also known as mCncc Iam) is an IAM component of M-CNCC. M-CNCC IAM contains M-CNCC IAM Ingress Gateway and M-CNCC IAM back-end microservices.
M-CNCC Kubernetes cluster	Kubernetes cluster hosting M-CNCC
MO	Managed Objects
MOS	My Oracle Support
mTLS	Mutual Transport Layer Security
NRF	Oracle Communications Cloud Native Core, Network Repository Function
OCI	Oracle Cloud Infrastructure
OCNADD	Oracle Communications Network Analytics Data Director
OCNF	Oracle Communications Network Function
OSDC	Oracle Software Delivery Cloud
OSO	Oracle Communications Operations Services Overlay
PROVGW	Provisioning Gateway
RBAC	Role Based Access Control
Release Stream	A release stream is a sequence of releases for a product available to the customer. Example of release streams are 24.1.x, 24.2.x, 24.3.x, 25.1.1xx, 25.1.2xx, 25.2.1xx, and so on.
REST API	Representational State Transfer Application Programming Interface
SAML	Security Assertion Markup Language
SBA	Service Based Architecture
SBI	Service Based Interface
SCP	Oracle Communications Cloud Native Core, Service Communication Proxy
SCSI	Single Cluster Single Instance
SEPP	Oracle Communications Cloud Native Core, Security Edge Protection Proxy
Site	Kubernetes Cluster
SSO	Single Sign On
TLS	Transport Layer Security
UDR	Oracle Communications Cloud Native Core, Unified Data Repository
UE	User Equipment
URI	Subscriber Location Function

What's New in This Guide

This section introduces the documentation updates for release 25.2.1xx.

Release 25.1.200 - G35643-02, November 2025

There are no updates in this release.

Release 25.1.200 - G35643-01, November 2025

- Added or updated the following rest specifications:
 - [Retrieve Specific User Information by Username](#)
 - [Update Existing User](#)

1

Introduction

This document provides information about how to configure the services and manageable objects in Oracle Communication Cloud Native Configuration Console using Representational State Transfer Application Program Interfaces (REST APIs).

1.1 Overview

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

The CNC Console has the following modules:

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

Oracle Cloud Infrastructure (OCI)

Oracle Cloud Infrastructure (OCI) is a set of complementary cloud services that enable you to build and run a range of applications and services in a highly available hosted environment. OCI provides high-performance compute capabilities (as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premises network.

OCI provides a single point platform for all Oracle products. Thus, deploying the CNC NFs on the OCI allows operational efficiency. In addition to this, the OCI also provides reduced Total Cost of Ownership (TCO), better performance, resource utilization, and persistence.

For more information, see [Oracle Cloud Infrastructure Documentation](#) and *Oracle Communications Cloud Native Core OCI Adaptor, NF Deployment on OCI Guide*.

Note

CNC Console IAM is not applicable for OCI deployment

1.2 Reference

Refer to the following documents for more information:

- *Oracle Communications Cloud Native Core, Service Communication Proxy User Guide*
- *Oracle Communications Cloud Native Core, Network Repository Function User Guide*
- *Oracle Communications Cloud Native Core, Policy User Guide*

- *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*
- *Oracle Communications Cloud Native Core, Binding Support Function User Guide*
- *Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide*
- *Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide*
- *Oracle Communications Cloud Native Core, Network Repository Function Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Service Communication Proxy Installation Guide*
- *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Binding Support Function Installation Guide*
- *Oracle Communications Cloud Native Core, Policy Installation Guide*
- *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation Guide*
- *Oracle Communications Cloud Native Core, Network Slice Selection Function Installation Guide*
- *Oracle Communications Networks Data Analytics Installation and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Certificate Management User Guide*
- *Oracle Communications Cloud Native Core, Certificate Management Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*
- *Oracle Communications Cloud Native Signaling Intelligence Hub User Guide*
- *Oracle Communications Cloud Native Signaling Intelligence Hub Installation and Troubleshooting Guide*
- *Oracle Communications Cloud Native Diameter Routing Agent User Guide*
- *Oracle Database Cloud Native Diameter Routing Agent Installation and Fault Recovery Guide*

2

REST Specifications for Non OCI Deployment

This chapter provides information about REST specifications used in Oracle Communications Cloud Native Configuration Console.

For HELM configurations, see *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*.

For the configurations using CNC Console, see *Oracle Communications Cloud Native Configuration Console User Guide*.

Note

The examples in this documentation use the cncc realm by default. To perform the same operations on the default realm, replace cncc with master in all URI paths and request payloads.

2.1 Generate access token

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

Type: POST

URI:

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

Table 2-1 Request Body Parameters

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

Sample URI:

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

Example curl command:

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

Example of the Request Body

The following is the example of the request body:

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

Example of the Response Body

The following is the example of the response body:

```
{  
    "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCIgOjAiAmlldUIiwia2lGMfag",  
    "expires_in": 60,  
    "refresh_expires_in": 1800,  
    "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCIgOjAiAmlldUIiwia2lkICIxM",  
    "not-before-policy": 0,  
    "session_state": "52dd8d7c-f8d9-4009-9c34-0262bb7d3722",  
    "scope": "email profile"  
}
```

Table 2-2 Supported Response Codes

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

2.2 Create a new user

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

Type: POST

URI:

POST /{realm}/users

Table 2-3 Request Body Parameters

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

Sample URI:

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

Example curl command:

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

Example of the Request Body

The following is the example of the request body:

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

Example of the Response Code

The following is the example of the response code:

201 Created

Table 2-4 Supported Response Codes

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

2.3 Get users

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

Type: GET

URI:

GET /{realm}/users

Sample URI:

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

Example curl command:

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

Example of the Response Body

The following is the example of the response body:

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

Table 2-5 Supported Response Codes

Code	Description
200 OK	Get users. Returns alist of users, filteredaccording to queryparameters.

Table 2-5 (Cont.) Supported Response Codes

Code	Description
401 Unauthorised	MissingAuthentication
404 Not Found	Realm not found

2.4 Retrieve Specific User Information by Username

CNC Console uses Get REST API to retrieve specific Uuer information by username.

Type: GET

URI:

GET /{realm}/users?username={username}

Sample URI:

http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users?username=user

Example curl command:

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

Example of the Response Body

The following is the example of the response body:

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

Table 2-6 Supported Response Codes

Code	Description
200 OK	If the user exists, return a non-empty array with user details. If the user does not exist, returns an empty array.
400 Bad Request	Invalid query parameter, malformed request, or missing required parameters.
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found

2.5 Get single user

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

Type: GET

URI:

GET /{realm}/users/{id}

Table 2-7 Request Path Parameter

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

Sample URI:

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

Example curl command:

```
curl --location --request GET 'http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users/754d6f6b-4ccb-44f1-abf1-00d717885dbe' \
--header 'Authorization: Bearer <token>'
```

Example of the Response Body

The following is the example of the response body:

```
{
  "id": "754d6f6b-4ccb-44f1-abf1-00d717885dbe",
  "createdTimestamp": 1661232491550,
  "username": "u1",
  "enabled": true,
  "totp": false,
```

```

    "emailVerified": false,
    "disableableCredentialTypes": [],
    "requiredActions": [],
    "notBefore": 0,
    "access": {
      "manageGroupMembership": true,
      "view": true,
      "mapRoles": true,
      "impersonate": true,
      "manage": true
    }
  }
}

```

Table 2-8 Supported Response Codes

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

2.6 Update Existing User

CNC Console uses Update existing user REST API to make updates to an existing user.

Type: PUT

URI:

PUT /{realm}/users/{id}

Sample URI:

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

Example curl command:

```

curl --location --request PUT 'http://{cncc-iam-ingress-extrenal-ip}:{cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users/7a2d3608-95b2-4a88-8efb-dad48e7778e2' \
-d '{
  "firstName": "CNCC",
  "lastName": "user",
  "email": "new@email.com",
  "enabled": true,
  "emailVerified": true
  "attributes": {
    "department": ["CNCC"]
  }
}'

```

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

Input Payload:

```
{  
  "firstName": "CNCC",  
  "lastName": "user",  
  "email": "new@email.com",  
  "enabled": true,  
  "emailVerified": true,  
  "attributes": {  
    "department": "CNCC"  
  }  
}
```

Example of the Response Body

The following is the example of the response body:

```
{  
  "id": "754d6f6b-4ccb-44f1-abf1-00d717885dbe",  
  "createdTimestamp": 1661232491550,  
  "username": "u1",  
  "email": "user@example.com",  
  "firstName": "CNCC",  
  "lastName": "User",  
  "enabled": true,  
  "totp": false,  
  "emailVerified": false,  
  "disableableCredentialTypes": [],  
  "requiredActions": [],  
  "notBefore": 0,  
  "access": {  
    "manageGroupMembership": true,  
    "view": true,  
    "mapRoles": true,  
    "impersonate": true,  
    "manage": true  
  }  
}
```

Table 2-9 Supported Response Codes

Code	Description
400 Bad Request	Invalid input data
401 Unauthorised	Missing Authentication
404 Not Found	Realm or user not found
409 Conflict	Email or username already exists

2.7 Delete the User

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

Type: DELETE

URI:

DELETE /{realm}/users/{id}

Table 2-10 Request Path Parameters

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

Sample URI:

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

Example curl command:

```
curl --location --request DELETE 'http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users/7a2d3608-95b2-4a88-8efb-dad48e7778e2' \--header 'Authorization: Bearer <token>'
```

Example of the Response Code

The following is the example of the response code:

204 No Content

Table 2-11 Supported Response Codes

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

2.8 Set up a New Password for the User

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

Type: PUT

URI:

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

Table 2-12 Request Body Parameters

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

Table 2-13 Request Path Parameter

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

Sample URI:

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

Example curl command:

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

Example of the Request Body

The following is the example of the request body:

```
{  
  "type": "password",  
  "value": "<Password Value>",  
  "temporary": false  
}
```

Example of the Response Code

The following is the example of the response code:

204 No Content

Table 2-14 Supported Response Codes

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

2.9 Reset the Temporary Password for a User

CNC Console uses Reset the Temporary Password for a User REST API to reset the temporary password for a user.

Type: PUT

URI:

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

Table 2-15 Request Body Parameters

Field Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
Type	String	M	Type
Value	String	M	Value of new password
Temporary	String	O	permanency

Table 2-16 Request Path Parameters

Field Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ID	String	M	ID of the user

Sample URI:

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

Example curl command:

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

Example of the Request Body

The following is the example of the request body:

```
{"value":"User123456!", "temporary":"false"}
```

Example of the Response Code

204 No Content

Table 2-17 Supported Response Codes

Code	Description
204 No Content	Reset the temporary password for the CNCC user (Requires payload).
401 Unauthorised	Missing Authentication
404 Not Found	Realm not found
404 Not Found	User not found
400 Bad Request	Invalid password

2.10 Get Realm Level Roles

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

Type: GET

URI:

GET /{realm}/roles

Sample URI:

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

Example curl command:

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

Example of the Response Body

The following is the example of the response body:

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

Table 2-18 Supported Response Codes

Code	Description
200 OK	Generate Get Realm Level Roles

2.11 Get Realm-level Role Mappings

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

Type: GET

URI:

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

Table 2-19 Request Path Parameters

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

Sample URI:

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

Example curl command:

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

Example of the Response Body

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

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

Table 2-20 Supported Response Codes

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

2.12 Add Realm-level Role Mappings to the User

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

Type:POST

URI:

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

Table 2-21 Request Body Parameters

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

Table 2-22 Request Path Parameter

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

Sample URI:

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

Example curl command:

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

Example of the Request Body

The following is the example of the request body:

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

Example of the Response Code

The following is the example of the response code:

204 No Content

Table 2-23 Supported Response Codes

Code	Description
204 No Content	Set up a newpassword for the CNCC user {Requires payload}.
401 Unauthorised	Missing Authentication

Table 2-23 (Cont.) Supported Response Codes

Code	Description
404 Not Found	Realm not found
404 Not Found	User not found
404 Not Found	Role not found

2.13 Get Realm-level Role Mappings for a specific user

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

Type:Get

URI:

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

Sample URI:

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

Example curl command:

```
curl --location --request GET 'http://${cncc-iam-ingress-extrenal-ip}:${cncc-iam-ingress-service-port}/cncc/auth/admin/realms/cncc/users/754d6f6b-4ccb-44f1-abf1-00d717885dbe/role-mappings/realm' \
--header 'Authorization: Bearer <token>'
```

Example of the Request Body

The following is the example of the request body:

```
[
  {
    "id": "fc5006e0-3927-4034-a01f-af70d779f1f8",
    "name": "ADMIN",
    "description": "Has access to all NF resources and can perform CRUD operations",
    "composite": true,
    "clientRole": false,
    "containerId": "cncc"
  },
  {
    "id": "503c9b40-b15e-4f88-891f-e71a495dd87f",
    "name": "Cluster1",
    "description": "Grants role to cluster1",
    "composite": false,
    "clientRole": false,
    "containerId": "cncc"
  }
]
```

```
    }, ...  
  ]
```

Table 2-24 Supported Response Codes

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

2.14 Remove All User Sessions Associated with the User

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

Type:POST

URI:

POST `/cnc/console/users/{id}/logout`

Table 2-25 Request Path Parameter

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

Sample URI:

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

Example curl command:

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

Example of the Response Code

The following is the example of the response code:

204 No Content

Table 2-26 Supported Response Codes

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

2.15 Get the Client details

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

Type: GET

URI:

GET/{realm}/clients

Sample URI:

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

Example curl command:

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

Example of the Response Body

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

```
[
  {
    "id": "b7fa17bd-135f-441b-a5f2-1ea4897e04fc",
    "clientId": "account",
    "name": "${client_account}",
    "rootUrl": "${authBaseUrl}",
    "baseUrl": "/realms/cncc/account/",
    "surrogateAuthRequired": false,
    "enabled": true,
    "alwaysDisplayInConsole": false,
    "clientAuthenticatorType": "client-secret",
    "redirectUris": [
      "/realms/cncc/account/"
    ],
    "webOrigins": [],
    "notBefore": 0,
    "bearerOnly": false,
    "consentRequired": false,
    "standardFlowEnabled": true,
```

```

    "implicitFlowEnabled": false,
    "directAccessGrantsEnabled": false,
    "serviceAccountsEnabled": false,
    "publicClient": false,
    "frontchannelLogout": false,
    "protocol": "openid-connect",
    "attributes": {},
    "authenticationFlowBindingOverrides": {},
    "fullScopeAllowed": false,
    "nodeReRegistrationTimeout": 0,
    "defaultClientScopes": [
        "web-origins",
        "roles",
        "profile",
        "email"
    ],
    "optionalClientScopes": [
        "address",
        "phone",
        "offline_access",
        "microprofile-jwt"
    ],
    "access": {
        "view": true,
        "configure": true,
        "manage": true
    }
},

```

2.16 Update the Client

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

Type: PUT

URI:

PUT/{realm}/clients/{id}

Table 2-27 Request Body Parameters

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

Table 2-28 Request Path Parameter

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

Sample URI:

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

Example curl command:

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

Example of the Request Body

The following is the example of the request body:

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

Example of the Response Code

The following is the example of the response code:

204 No Content

Table 2-29 Supported Response Codes

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

3

REST Specification for OCI Deployment

This section provides information about REST specifications used in Oracle Communications Cloud Native Configuration Console while deployed in Oracle Cloud Infrastructure (OCI).

3.1 Generate Access Token

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

Type: POST

URI:

/oauth2/v1/token

Sample URI:

https://<oci-iam-domain-url>/v1/token

Example curl command:

```
curl --location 'https://<oci-iam-domain-url>/oauth2/v1/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Authorization: Basic <Base64 encoding of client credentials in the
format "clientId:clientSceret">' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'username=testuser' \
--data-urlencode 'password=*****' \
--data-urlencode 'scope=urn:opc:idm:__myscopes__ offline_access' -s
```

Example of the Response Body

The following is the example of the response body:

```
200 OK
{
  "access_token":
  "eyJhbGciOiJSUzI1NiIsI....._LcCZYwDQJTTloj2PJ8y1Wj09l2Q",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

3.2 Create a New User

CNC Console uses Create a new user REST API to create a new user.

Type: POST

URI:

/admin/v1/Users

Sample URI:

https://<oci-iam-domain-url>/admin/v1/Users

Example curl command:

```
curl --location 'https://<oci-iam-domain-url>/admin/v1/Users' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsI....._LcCZYWDQJJTloj2PJ8y1Wj09l2Q' \
--header 'Content-Type: application/json' \
--data-raw '{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "name": {
    "givenName": "user",
    "familyName": "test"
  },
  "userName": "user@oracle.com",
  "emails": [
    {
      "value": "user@oracle.com",
      "type": "work",
      "primary": true
    }
  ]
}'
```

Example of the Response Body

The following is the example of the response body:

```
201 OK
{
  "idcsCreatedBy": {
    "type": "User",
    "display": "Test User",
    "value": "5427ea861e2c479895b8375b19c6827a",
    "ocid":
      "ocid1.user.oc1..aaaaaaaaxszu5pi3iudjfa2tkkz3onobfnk3ctuqqyg3zsa7gx4sohikrboq"
  },
  "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/5427ea861e2c479895b8375b19c6827a"
},
  "id": "de18f42c3eeb474d98853574b1927200",
  "urn:ietf:params:scim:schemas:oracle:idcs:extension:user:User": {
    "isFederatedUser": false,
    "preferredUiLandingPage": "MyApps"
  },
}
```

```

    "meta": {
      "created": "2024-01-23T11:01:02.009Z",
      "lastModified": "2024-01-23T11:01:02.009Z",
      "version": "7842e880cafc44d49f30cda44436cfc2",
      "resourceType": "User",
      "location": "https://<oci-iam-domain-url>:443/admin/v1/Users/
de18f42c3eeb474d98853574b1927200"
    },
    "active": true,
    "displayName": "user121 dummy",
    "idcsLastModifiedBy": {
      "value": "5427ea861e2c479895b8375b19c6827a",
      "display": "Test User",
      "ocid":
"ocid1.user.oc1..aaaaaaaaxszu5pi3iudjfa2tkkz3onobfnk3ctuqqyg3zsa7gx4sohikrboq"
    },
    "type": "User",
    "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
5427ea861e2c479895b8375b19c6827a"
  },
  "name": {
    "givenName": "user121",
    "familyName": "dummy",
    "formatted": "user121 dummy"
  },
  "urn:ietf:params:scim:schemas:oracle:idcs:extension:userState:User": {
    "locked": {
      "on": false
    }
  },
  "ocid": "ocid1.user.oc1...gwvwq",
  "userName": "user121",
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User",
    "urn:ietf:params:scim:schemas:oracle:idcs:extension:userState:User",
    "urn:ietf:params:scim:schemas:oracle:idcs:extension:user:User"
  ],
  "domainOcid": "ocid1.domain.oc1...j6kgka",
  "compartmentOcid": "ocid1.tenancy.oc1...pojwe4wx34a",
  "tenancyOcid": "ocid1.tenancy.oc1...wx34a",
  "urn:ietf:params:scim:schemas:oracle:idcs:extension:capabilities:User": {
    "canUseApiKeys": true,
    "canUseAuthTokens": true,
    "canUseConsolePassword": true,
    "canUseCustomerSecretKeys": true,
    "canUseOAuth2ClientCredentials": true,
    "canUseSmtptCredentials": true,
    "canUseDbCredentials": true
  }
}

```

3.3 Get All User

CNC Console uses Get all user REST API to get all users.

Type: GET

URI:

/admin/v1/Users

Sample URI:

https://<oci-iam-domain-url>/admin/v1/Users

Example curl command:

```
curl --location 'https://<oci-iam-domain-url>/admin/v1/Users' \
--header 'Authorization: Bearer
'eyJhbGciOiJIUzI1NiIsI....._LcCZYwDQJJTloj2PJ8y1Wj09l2Q'
```

Example of the Response Body

The following is the example of the response body:

```
201 OK
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 36,
  "Resources": [
    {
      "urn:ietf:params:scim:schemas:oracle:ids:extension:user:User": {
        "isFederatedUser": false,
        "preferredUiLandingPage": "MyApps"
      },
      "idsCreatedBy": {
        "type": "App",
        "display": "idsjob",
        "value": "4ed77d9256654268aa286ec1a959dbe2",
        "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/4ed77d9256654268aa286ec1a959dbe2"
      },
      "id": "a07ebe1f5b5543608803561b97265fcd",
      "meta": {
        "created": "2024-01-18T05:39:01.978Z",
        "lastModified": "2024-01-18T05:39:01.978Z",
        "version": "186f6d8f4364492ab5c79659d2c33263",
        "resourceType": "User",
        "location": "https://<oci-iam-domain-url>:443/admin/v1/Users/a07ebe1f5b5543608803561b97265fcd"
      },
      "active": true,
      "displayName": "CSV User7",
      "idsLastModifiedBy": {
        "value": "4ed77d9256654268aa286ec1a959dbe2",
        "display": "idsjob",
        "type": "App",
        "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/"
      }
    }
  ]
}
```

```

4ed77d9256654268aa286ec1a959dbe2"
    },
    "name": {
      "givenName": "CSV",
      "familyName": "User7",
      "formatted": "CSV User7"
    },
    "urn:ietf:params:scim:schemas:oracle:ids:extension:userState:User": {
      "locked": {
        "on": false
      }
    },
    "ocid":
"ocid1.user.oc1..aaaaaaaarql7rbnj4s7ayjiykve7ozjoxfkp7dub7gpwu6vpc13ysm4he4ha"
,
    "userName": "CSV7",
    "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:User",
      "urn:ietf:params:scim:schemas:oracle:ids:extension:userState:User",
      "urn:ietf:params:scim:schemas:oracle:ids:extension:user:User"
    ],
    "domainOcid":
"ocid1.domain.oc1..aaaaaaaatzwhla6vhqwnvj6eiaoksjwqe2sxxwzls6zljr3ttcs5mj6kgk
a",
    "compartmentOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
    "tenancyOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
    "urn:ietf:params:scim:schemas:oracle:ids:extension:capabilities:User": {
      "canUseApiKeys": true,
      "canUseAuthTokens": true,
      "canUseConsolePassword": true,
      "canUseCustomerSecretKeys": true,
      "canUseOAuth2ClientCredentials": true,
      "canUseSmtppCredentials": true,
      "canUseDbCredentials": true
    }
  },
  .
  .
  .
}

```

3.4 Get Specific User

CNC Console uses Get all user REST API to get a specific user.

Type: GET

URI:

/admin/v1/Users/{{userid}}

Sample URI:

https://<oci-iam-domain-url>/admin/v1/Users/{{userid}}

Sample user format:

```
{
  "id": 12...x3,
  ...
  "ocid": "ocid1...op",
  "userName": "testuser"
  .
  .
  .
}
```

For {{userid}} field, "id" as well as "ocid" can be used.

Example curl command:

```
curl --location 'https://<oci-iam-domain-url>/admin/v1/Users/
ocid1.user.oc1...jrdq' \
--header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsI....._LcCZYWDQJTTloj2PJ8y1Wj09l2Q'
```

Example of the Response Body

The following is the example of the response body:

```
200 OK{
  "idcsCreatedBy": {
    "type": "User",
    "display": "User1",
    "value": "411bdad4c26044aaa8c8cab88da48e86",
    "ocid":
"ocid1.user.oc1..aaaaaaa6woa5vcnxt05mcxafx5isaat77tqdnj62iulxxeiycwu4w7aq"
  },
  "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
411bdad4c26044aaa8c8cab88da48e86"
},
  "id": "30857d89710942e2b5bd67e37d7c0c15",
  "urn:ietf:params:scim:schemas:oracle:idcs:extension:user:User": {
    "isFederatedUser": false
  },
  "meta": {
    "created": "2023-12-15T10:09:05.487Z",
    "lastModified": "2024-01-11T07:11:23.485Z",
    "version": "75a47233965648079c0b09aadb1c4d64",
    "resourceType": "User",
    "location": "https://<oci-iam-domain-url>:443/admin/v1/Users/
ocid1.user.oc1..aaaaaaaafnvamuekbzxjzyg3fd2b203eimcw3hncypjnfudiyflfid3jrdq"
  },
}
```

```

    "active": true,
    "displayName": "Karan Gupta",
    "urn:ietf:params:scim:schemas:oracle:idcs:extension:mfa:User": {
      "preferredDevice": {
        "value": "620a0cde1f6e42f38fca609a6097fbfe",
        "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Devices/620a0cde1f6e42f38fca609a6097fbfe"
      },
      "loginAttempts": 0,
      "mfaStatus": "ENROLLED",
      "preferredAuthenticationFactor": "PUSH"
    },
    "idcsLastModifiedBy": {
      "value": "af2d1df6c13f493b9f3131baae2e8638",
      "display": "idcssso",
      "ocid": "ocid1.domainapp.oc1.iad.amaaaaaasvq5r3yarymeans3hx2mz34nkypz3pan477oeiithvx3ax34asaa",
      "type": "App",
      "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/af2d1df6c13f493b9f3131baae2e8638"
    },
    "name": {
      "givenName": "Karan",
      "familyName": "Gupta",
      "formatted": "Karan Gupta"
    },
    "urn:ietf:params:scim:schemas:oracle:idcs:extension:userState:User": {
      "locked": {
        "on": false
      }
    },
    "ocid": "ocid1.user.oc1..aaaaaaaafnvamuekbzxjzyg3fd2b2o3eimcw3hnyccypjnfudiyflfid3jrdq"
  },
  "userName": "karan",
  "urn:ietf:params:scim:schemas:oracle:idcs:extension:OCITags": {
    "definedTags": [
      {
        "value": "corp/karan.k.gupta@oracle.com",
        "key": "CreatedBy",
        "namespace": "Oracle-Tags"
      },
      {
        "value": "2023-12-15T10:09:05.454Z",
        "key": "CreatedOn",
        "namespace": "Oracle-Tags"
      }
    ]
  },
  "emails": [
    {
      "secondary": false,
      "verified": true,
      "type": "recovery",
      "primary": false,

```

```

        "value": "karan.k.gupta@oracle.com"
      },
      {
        "secondary": false,
        "verified": true,
        "type": "work",
        "primary": true,
        "value": "karan.k.gupta@oracle.com"
      }
    ],
    "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:User",
      "urn:ietf:params:scim:schemas:oracle:ids:extension:userState:User",
      "urn:ietf:params:scim:schemas:oracle:ids:extension:OCITags",
      "urn:ietf:params:scim:schemas:oracle:ids:extension:mfa:User",
      "urn:ietf:params:scim:schemas:oracle:ids:extension:user:User"
    ],
    "domainOcid":
      "ocid1.domain.oc1..aaaaaaaatzwhla6vhqwnvj6eiaoksjqwe2sxxwzls6zlj3ttcs5mj6k9ka",
    "compartmentOcid":
      "ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx34a",
    "tenancyOcid":
      "ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx34a",
    "urn:ietf:params:scim:schemas:oracle:ids:extension:capabilities:User": {
      "canUseApiKeys": true,
      "canUseAuthTokens": true,
      "canUseConsolePassword": true,
      "canUseCustomerSecretKeys": true,
      "canUseOAuth2ClientCredentials": true,
      "canUseSmtPcredentials": true,
      "canUseDbCredentials": true
    }
  }
}

```

3.5 Delete User

CNC Console uses Delete User REST API to delete user.

Type: DELETE

URI:

/admin/v1/Users/{userid}

Sample URI:

https://<oci-iam-domain-url>/admin/v1/Users/{userid}

Sample user format:

```
{
  "id": 12...x3,
  ...
  "ocid": "ocid1...op",
  "userName": "testuser"
  .
  .
  .
}
```

For {{userid}} field, "id" as well as "ocid" can be used.

Example curl command:

```
curl --location --request DELETE 'https://<oci-iam-domain-url>/admin/v1/Users/
ocid1.user.oc1...fs6tq' \
--header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsI....._LcCZYWDQJJTloj2PJ8y1Wj09l2Q' \
--header 'Content-Type: application/json'
```

Example of the Response Body

The following is the example of the response body:

204 No Content

3.6 Set up a New Password for the User

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

Type: GET

URI:

/admin/v1/UserPasswordChanger/{{userid}}

Sample URI:

https://<oci-iam-domain-url>/admin/v1/UserPasswordChanger/{{userid}}

Sample user format:

```
{
  "id": 12...x3,
  ...
  "ocid": "ocid1...op",
  "userName": "testuser"
  .
  .
  .
}
```

```
.
}
```

For {{userid}} field, "id" as well as "ocid" can be used.

Example curl command:

```
curl --location --request PUT 'https://<oci-iam-domain-url>/admin/v1/
UserPasswordChanger/ocid1.user.oc1...vpcta' \
--header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsI....._LcCZYwDQJJTloj2PJ8y1Wj09l2Q' \
--header 'Content-Type: application/json' \
--data-raw '{
  "password": "Welcome@123Bro",
  "schemas": [
    "urn:ietf:params:scim:schemas:oracle:ids:UserPasswordChanger"
  ]
}'
```

Example of the Response Body

The following is the example of the response body:

```
200 OK
{
  "schemas": [
    "urn:ietf:params:scim:schemas:oracle:ids:UserPasswordChanger"
  ],
  "id": "023aa77a7ccd419db579dc6707c43e03",
  "meta": {
    "resourceType": "UserPasswordChanger",
    "location": "https://<oci-iam-domain-url>:443/admin/v1/
UserPasswordChanger/
ocid1.user.oc1..aaaaaaa7hz5lh4a64aekqndcvolrhscj25udrz7gux44k17pckdlgvpcta"
  },
  "domainOcid":
"ocid1.domain.oc1..aaaaaaaatzwh1a6vhqwnvj6eiaoksjwtq2sxxwzls6zljr3ttcs5mj6kgk
a",
  "compartmentOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
  "tenancyOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
  "ocid":
"ocid1.user.oc1..aaaaaaa7hz5lh4a64aekqndcvolrhscj25udrz7gux44k17pckdlgvpcta"
}

400 Bad Request
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error",
    "urn:ietf:params:scim:api:oracle:ids:extension:messages:Error"
  ],
```

```

    "detail": "The password does not meet the criteria of the password policy
because: [The password must have at least 12 characters.]",
    "status": "400",
    "urn:ietf:params:scim:api:oracle:ids:extension:messages:Error": {
      "messageId": "error.identity.passwordmgmt.pwdpolicyViolation"
    }
  }
}

```

3.7 Get all groups of the domain

CNC Console uses Get all groups of the domain REST API to get all groups of the domain.

Type: GET

URI:

/admin/v1/Groups

Sample URI:

`https://<oci-iam-domain-url>/admin/v1/Groups`

Example curl command:

```

curl --location 'https://<oci-iam-domain-url>/admin/v1/Groups' \
--header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsI....._LcCZYWDQJJTloj2PJ8y1Wj09l2Q'

```

Example of the Response Body

The following is the example of the response body:

```

200 OK
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 58,
  "Resources": [
    {
      "displayName": "SCP_WRITE",
      "idcsLastModifiedBy": {
        "type": "User",
        "value": "fc0f9952563e4b4399f4869fc066673d",
        "display": "User1",
        "ocid":
"ocid1.user.oc1..aaaaaaa7ieth5a2q72o52uu56cx3xwigqldjpn dp63zt7jo4a7hhhcnyvpq"
      },
      "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
fc0f9952563e4b4399f4869fc066673d"
    },
    "idcsCreatedBy": {

```

```

        "type": "User",
        "ocid":
"ocid1.user.oc1..aaaaaaa7ieth5a2q72o52uu56cx3xwigqljdpndp63zt7jo4a7hhhcnyvpq"
,
        "display": "User1",
        "value": "fc0f9952563e4b4399f4869fc066673d",
        "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
fc0f9952563e4b4399f4869fc066673d"
    },
    "urn:ietf:params:scim:schemas:oracle:ids:extension:group:Group": {
        "description": "Read and Write access to SCP Resource"
    },
    "meta": {
        "version": "418190f8f2c64e4fa380f5d2673b9f37",
        "created": "2023-11-14T09:05:39.001Z",
        "lastModified": "2023-11-21T03:53:00.352Z",
        "resourceType": "Group",
        "location": "https://<oci-iam-domain-url>:443/admin/v1/Groups/
d29818fc6a1d4be9883699766a8f026c"
    },
    "id": "d29818fc6a1d4be9883699766a8f026c",
    "ocid":
"ocid1.group.oc1..aaaaaaaahwdyfqqvz42tszory5nkz4kuha6iuwah26j4pgtuyhic3gssq2jq
",
    "urn:ietf:params:scim:schemas:oracle:ids:extension:OCITags": {
        "definedTags": [
            {
                "key": "CreatedBy",
                "namespace": "Oracle-Tags",
                "value": "corp/shreyas.b@oracle.com"
            },
            {
                "key": "CreatedOn",
                "namespace": "Oracle-Tags",
                "value": "2023-11-14T09:05:38.977Z"
            }
        ]
    },
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:Group",
        "urn:ietf:params:scim:schemas:oracle:ids:extension:OCITags",

        "urn:ietf:params:scim:schemas:oracle:ids:extension:group:Group"
    ],
    "domainOcid":
"ocid1.domain.oc1..aaaaaaaatzwhla6vhqwnvj6eiaoksjqwe2sxxwzls6zljr3ttcs5mj6kgk
a",
    "compartmentOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
    "tenancyOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",

    "urn:ietf:params:scim:schemas:oracle:ids:extension:dynamic:Group": {
        "membershipType": "static"
    }

```

```

    }
  },
  {
    "displayName": "SCP_WRITE",
    "idcsLastModifiedBy": {
      "type": "User",
      "value": "5427ea861e2c479895b8375b19c6827a",
      "display": "Test User",
      "ocid":
"ocid1.user.oc1..aaaaaaaaxszu5pi3iudjfa2tkkz3onobfnk3ctuqqyg3zsa7gx4sohikrboq"
    },
    "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
5427ea861e2c479895b8375b19c6827a"
  },
  "idcsCreatedBy": {
    "type": "User",
    "ocid":
"ocid1.user.oc1..aaaaaaaaxszu5pi3iudjfa2tkkz3onobfnk3ctuqqyg3zsa7gx4sohikrboq"
  },
  "display": "Test User",
  "value": "5427ea861e2c479895b8375b19c6827a",
  "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
5427ea861e2c479895b8375b19c6827a"
},
"urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group": {
  "description": "Example Description"
},
"meta": {
  "version": "1d57c48a69c1485cbfa732d0f6c6d5b5",
  "created": "2024-01-16T05:58:21.652Z",
  "lastModified": "2024-01-16T05:58:21.652Z",
  "resourceType": "Group",
  "location": "https://<oci-iam-domain-url>:443/admin/v1/Groups/
0a71625b839a41d4a70814bd336e9580"
},
"id": "0a71625b839a41d4a70814bd336e9580",
"ocid":
"ocid1.group.oc1..aaaaaaaazu4nl2jls15iavvnvjszbwrldmyqn4fo5n7hbyng6mooaamsjynfq"
",
  "externalId": "123456",
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group",

    "urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group"
  ],
  "domainOcid":
"ocid1.domain.oc1..aaaaaaaatzwhla6vhqwnvj6eiaoksjwqe2sxxwzls6zljr3ttcs5mj6kgk
a",
  "compartmentOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngieby6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
  "tenancyOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngieby6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",

  "urn:ietf:params:scim:schemas:oracle:idcs:extension:dynamic:Group": {

```

```

        "membershipType": "static"
      },
    ],
  },
}

400 Bad Request
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:Error",
    "urn:ietf:params:scim:api:oracle:ids:extension:messages:Error"
  ],
  "detail": "The password does not meet the criteria of the password policy
because: [The password must have at least 12 characters.]",
  "status": "400",
  "urn:ietf:params:scim:api:oracle:ids:extension:messages:Error": {
    "messageId": "error.identity.passwordmgmt.pwdpolicyViolation"
  }
}

```

3.8 Get all groups of a specific user

CNC Console uses Get all groups of a specific user REST API to get all groups of a specific user.

Type: GET

URI:

/admin/v1/Groups?filter=members[type eq "User" and value eq "{{userid}}"]

Sample URI:

https://<oci-iam-domain-url>/admin/v1/Groups?filter=members[type eq "User" and value eq "{{userid}}"]

Sample user format:

```

{
  "id": 12...x3,
  ...
  "ocid": "ocid1...op",
  "userName": "testuser"
  .
  .
  .
}

```

For {{userid}} field, "id" as well as "ocid" can be used.

Example curl command:

Replace <USER-ID> with required user's id .

```
curl --location --globoff 'https://<oci-iam-domain-url>/admin/v1/Groups?
filter=members[type%20eq%20%22User%22%20and%20value%20eq%20%22<USER-ID>%22]'
\--header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsI....._LcCZYwDQJJTloj2PJ8y1Wj09l2Q'
```

Example of the Response Body

The following is the example of the response body:

```
200 OK{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 3,
  "Resources": [
    {
      "displayName": "Domain Administrators",
      "idcsLastModifiedBy": {
        "type": "App",
        "value": "6923eddd11d1457084f550273b51027a",
        "display": "idcssm",
        "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/
6923eddd11d1457084f550273b51027a"
      },
      "idcsCreatedBy": {
        "type": "App",
        "display": "idcssm",
        "value": "6923eddd11d1457084f550273b51027a",
        "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/
6923eddd11d1457084f550273b51027a"
      },
      "urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group":
    {
      "description": "Domain Administrators"
    },
    "meta": {
      "version": "a9d7f9a9e7114c2994fe729f0506739a",
      "created": "2023-10-06T05:23:58.569Z",
      "lastModified": "2023-10-06T05:23:58.569Z",
      "resourceType": "Group",
      "location": "https://<oci-iam-domain-url>:443/admin/v1/Groups/
2c715a2f00b44e3fab0dced1c5798d84"
    },
    "id": "2c715a2f00b44e3fab0dced1c5798d84",
    "ocid":
"ocid1.group.oc1..aaaaaaaenl5su2ejqkzonymjdcrtt52nvgfk53zgggjp2vfja5lya742dqng
",
    "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:Group",
      "urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group"
    ]
  ]
}
```

```

        ],
        "domainOcid":
"ocid1.domain.oc1..aaaaaaaatzwhla6vhqwnvj6eiaoksjwtq2sxxwzls6zljr3ttcs5mj6kgk
a",
        "compartmentOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
        "tenancyOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
        "urn:ietf:params:scim:schemas:oracle:ids:extension:dynamic:Group": {
            "membershipType": "static"
        }
    },
    {
        "displayName": "SCP_READ",
        "idsLastModifiedBy": {
            "type": "App",
            "value": "4ed77d9256654268aa286ec1a959dbe2",
            "display": "idsjob",
            "ocid":
"ocid1.user.oc1..aaaaaaa7ieth5a2q72o52uu56cx3xwigqldjpndp63zt7jo4a7hhhcnyvpq"
,
            "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/
4ed77d9256654268aa286ec1a959dbe2"
        },
        "idsCreatedBy": {
            "type": "User",
            "ocid":
"ocid1.user.oc1..aaaaaaa7ieth5a2q72o52uu56cx3xwigqldjpndp63zt7jo4a7hhhcnyvpq"
,
            "display": "Shreyas Rao B",
            "value": "fc0f9952563e4b4399f4869fc066673d",
            "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
fc0f9952563e4b4399f4869fc066673d"
        },
        "urn:ietf:params:scim:schemas:oracle:ids:extension:group:Group":
{
            "description": "Has access to only SCP resources and can only
perform READ Managed Objects of SCP."
        },
        "meta": {
            "version": "8a1f0617369a444da296ef6ce2b73a9e",
            "created": "2023-11-14T09:05:28.413Z",
            "lastModified": "2024-01-25T03:56:30.173Z",
            "resourceType": "Group",
            "location": "https://<oci-iam-domain-url>:443/admin/v1/Groups/
5d315d352e8c4af38e590b71ed6d2699"
        },
        "id": "5d315d352e8c4af38e590b71ed6d2699",
        "ocid":
"ocid1.group.oc1..aaaaaaaava7e65qhum3tatqzzk43tqyb5f7txwb56ukbhtlxf5neb54yy4oq
",
        "urn:ietf:params:scim:schemas:oracle:ids:extension:OCITags": {
            "definedTags": [

```

```

        {
            "key": "CreatedBy",
            "namespace": "Oracle-Tags",
            "value": "corp/shreyas.b@oracle.com"
        },
        {
            "key": "CreatedOn",
            "namespace": "Oracle-Tags",
            "value": "2023-11-14T09:05:28.391Z"
        }
    ],
    },
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:Group",
        "urn:ietf:params:scim:schemas:oracle:ids:extension:OCITags",
        "urn:ietf:params:scim:schemas:oracle:ids:extension:group:Group"
    ],
    "domainOcid":
    "ocid1.domain.oc1..aaaaaaaatzwhla6vhqwnvj6eiaoksjqwe2sxxwzls6zljr3ttcs5mj6kgk
a",
    "compartmentOcid":
    "ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
    "tenancyOcid":
    "ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
    "urn:ietf:params:scim:schemas:oracle:ids:extension:dynamic:Group": {
        "membershipType": "static"
    }
},
{
    "displayName": "SCP_WRITE",
    "idsLastModifiedBy": {
        "type": "App",
        "value": "4ed77d9256654268aa286ec1a959dbe2",
        "display": "idcsjob",
        "ocid":
        "ocid1.user.oc1..aaaaaaa7ieth5a2q72o52uu56cx3xwigqljdpndp63zt7jo4a7hhhcnyvpq"
        ,
        "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/
4ed77d9256654268aa286ec1a959dbe2"
    },
    "idsCreatedBy": {
        "type": "User",
        "ocid":
        "ocid1.user.oc1..aaaaaaa7ieth5a2q72o52uu56cx3xwigqljdpndp63zt7jo4a7hhhcnyvpq"
        ,
        "display": "Shreyas Rao B",
        "value": "fc0f9952563e4b4399f4869fc066673d",
        "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
fc0f9952563e4b4399f4869fc066673d"
    },
    "urn:ietf:params:scim:schemas:oracle:ids:extension:group:Group":
    {

```

```

        "description": "Has access to only SCP resources and can
perform CRUD operations on Managed Objects."
    },
    "meta": {
        "version": "f4fcceb2a4014355a0a099ea22f0f2da",
        "created": "2023-11-14T09:05:39.001Z",
        "lastModified": "2024-01-25T03:56:30.043Z",
        "resourceType": "Group",
        "location": "https://<oci-iam-domain-url>:443/admin/v1/Groups/
d29818fc6a1d4be9883699766a8f026c"
    },
    "id": "d29818fc6a1d4be9883699766a8f026c",
    "ocid":
"ocid1.group.oc1..aaaaaaaahwdyfqkvz42tszory5nkz4kuha6iuwah26j4pgtuyhic3gssq2jq
",
    "urn:ietf:params:scim:schemas:oracle:ids:extension:OCITags": {
        "definedTags": [
            {
                "key": "CreatedBy",
                "namespace": "Oracle-Tags",
                "value": "corp/shreyas.b@oracle.com"
            },
            {
                "key": "CreatedOn",
                "namespace": "Oracle-Tags",
                "value": "2023-11-14T09:05:38.977Z"
            }
        ]
    },
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:Group",
        "urn:ietf:params:scim:schemas:oracle:ids:extension:OCITags",

        "urn:ietf:params:scim:schemas:oracle:ids:extension:group:Group"
    ],
    "domainOcid":
"ocid1.domain.oc1..aaaaaaaatzwhla6vhqwnvj6eiaoksjwtq2sxxwzls6zljr3ttcs5mj6kgk
a",
    "compartmentOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
    "tenancyOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
    "urn:ietf:params:scim:schemas:oracle:ids:extension:dynamic:Group": {
        "membershipType": "static"
    }
},
"startIndex": 1,
"itemsPerPage": 50
}

```

3.9 Create a group

CNC Console uses Create a group REST API to create a group.

Type: POST

URI:

/admin/v1/Groups

Sample URI:

https://<oci-iam-domain-url>/admin/v1/Groups

Sample user format:

```
{
  "id": 12...x3,
  ...
  "ocid": "ocid1...op",
  "userName": "testuser"
}
```

For {{userid}} field, "id" as well as "ocid" can be used.

Example curl command:

```
curl --location --globoff 'https://<oci-iam-domain-url>/admin/v1/Groups?
filter=members[type%20eq%20%22User%22%20and%20value%20eq%20%22ocid1.user.oc1..
.q%22]' \
--header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsI....._LcCZYwDQJJTloj2PJ8y1Wj09l2Q'
```

Example of the Response Body

The following is the example of the response body:

```
200 OK{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 3,
  "Resources": [
    {
      "displayName": "Domain_Administrators",
      "idcsLastModifiedBy": {
        "type": "App",
        "value": "6923eddd11d1457084f550273b51027a",
        "display": "idcssm",
        "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/"
      }
    }
  ]
}
```

```

6923eddd11d1457084f550273b51027a"
    },
    "idcsCreatedBy": {
      "type": "App",
      "display": "idcssm",
      "value": "6923eddd11d1457084f550273b51027a",
      "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/
6923eddd11d1457084f550273b51027a"
    },
    "urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group":
  {
    "description": "Domain Administrators"
  },
  "meta": {
    "version": "a9d7f9a9e7114c2994fe729f0506739a",
    "created": "2023-10-06T05:23:58.569Z",
    "lastModified": "2023-10-06T05:23:58.569Z",
    "resourceType": "Group",
    "location": "https://<oci-iam-domain-url>:443/admin/v1/Groups/
2c715a2f00b44e3fab0dced1c5798d84"
  },
  "id": "2c715a2f00b44e3fab0dced1c5798d84",
  "ocid":
"ocid1.group.oc1..aaaaaaaenl5su2ejqkzonymjdcrtt52nvgfk53zgggjp2vfja5lya742dqnq
",
    "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:Group",

      "urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group"
    ],
    "domainOcid":
"ocid1.domain.oc1..aaaaaaaatzwhla6vhqwnvj6eiaoksjwtq2sxxwzls6zljr3ttcs5mj6kgk
a",
    "compartmentOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngieby6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
    "tenancyOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngieby6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",

    "urn:ietf:params:scim:schemas:oracle:idcs:extension:dynamic:Group": {
      "membershipType": "static"
    }
  },
  {
    "displayName": "SCP_READ",
    "idcsLastModifiedBy": {
      "type": "App",
      "value": "4ed77d9256654268aa286ec1a959dbe2",
      "display": "idcsjob",
      "ocid":
"ocid1.user.oc1..aaaaaaa7ieth5a2q72o52uu56cx3xwigqldjpn63zt7jo4a7hhhcnyvpq"
,
      "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/
4ed77d9256654268aa286ec1a959dbe2"
    },

```

```

        "idcsCreatedBy": {
            "type": "User",
            "ocid":
"ocid1.user.oc1..aaaaaaa7ieth5a2q72o52uu56cx3xwigqldjpndp63zt7jo4a7hhhcnyvpq"
,
            "display": "Shreyas Rao B",
            "value": "fc0f9952563e4b4399f4869fc066673d",
            "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
fc0f9952563e4b4399f4869fc066673d"
        },
        "urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group":
{
    "description": "Has access to only SCP resources and can only
perform READ Managed Objects of SCP."
},
    "meta": {
        "version": "8a1f0617369a444da296ef6ce2b73a9e",
        "created": "2023-11-14T09:05:28.413Z",
        "lastModified": "2024-01-25T03:56:30.173Z",
        "resourceType": "Group",
        "location": "https://<oci-iam-domain-url>:443/admin/v1/Groups/
5d315d352e8c4af38e590b71ed6d2699"
    },
    "id": "5d315d352e8c4af38e590b71ed6d2699",
    "ocid":
"ocid1.group.oc1..aaaaaaaava7e65qhum3tatqzzk43tqyb5f7txwb56ukbhtlxf5neb54yy4oq"
,
    "urn:ietf:params:scim:schemas:oracle:idcs:extension:OCITags": {
        "definedTags": [
            {
                "key": "CreatedBy",
                "namespace": "Oracle-Tags",
                "value": "corp/shreyas.b@oracle.com"
            },
            {
                "key": "CreatedOn",
                "namespace": "Oracle-Tags",
                "value": "2023-11-14T09:05:28.391Z"
            }
        ]
    },
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:Group",
        "urn:ietf:params:scim:schemas:oracle:idcs:extension:OCITags",

"urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group"
    ],
    "domainOcid":
"ocid1.domain.oc1..aaaaaaaatzwhla6vhqwnvj6eiaoksjwqe2sxxwzls6zljr3ttcs5mj6kgk
a",
    "compartmentOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",
    "tenancyOcid":
"ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
4a",

```

```

"urn:ietf:params:scim:schemas:oracle:ids:extension:dynamic:Group": {
  "membershipType": "static"
},
{
  "displayName": "SCP_WRITE",
  "idcsLastModifiedBy": {
    "type": "App",
    "value": "4ed77d9256654268aa286ec1a959dbe2",
    "display": "idcsjob",
    "ocid":
"ocid1.user.oc1..aaaaaaa7ieth5a2q72o52uu56cx3xwigqldjpndp63zt7jo4a7hhhcnyvpq"
,
    "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Apps/
4ed77d9256654268aa286ec1a959dbe2"
  },
  "idcsCreatedBy": {
    "type": "User",
    "ocid":
"ocid1.user.oc1..aaaaaaa7ieth5a2q72o52uu56cx3xwigqldjpndp63zt7jo4a7hhhcnyvpq"
,
    "display": "Shreyas Rao B",
    "value": "fc0f9952563e4b4399f4869fc066673d",
    "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
fc0f9952563e4b4399f4869fc066673d"
  },
  "urn:ietf:params:scim:schemas:oracle:ids:extension:group:Group":
{
  "description": "Has access to only SCP resources and can
perform CRUD operations on Managed Objects."
},
  "meta": {
    "version": "f4fcceb2a4014355a0a099ea22f0f2da",
    "created": "2023-11-14T09:05:39.001Z",
    "lastModified": "2024-01-25T03:56:30.043Z",
    "resourceType": "Group",
    "location": "https://<oci-iam-domain-url>:443/admin/v1/Groups/
d29818fc6a1d4be9883699766a8f026c"
  },
  "id": "d29818fc6a1d4be9883699766a8f026c",
  "ocid":
"ocid1.group.oc1..aaaaaaaahwdyfqqkvz42tszory5nkz4kuha6iuwah26j4pgtuyhic3gssq2jq
",
  "urn:ietf:params:scim:schemas:oracle:ids:extension:OCITags": {
    "definedTags": [
      {
        "key": "CreatedBy",
        "namespace": "Oracle-Tags",
        "value": "corp/shreyas.b@oracle.com"
      },
      {
        "key": "CreatedOn",
        "namespace": "Oracle-Tags",
        "value": "2023-11-14T09:05:38.977Z"
      }
    ]
  }
}

```

```

    ],
    "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:Group",
      "urn:ietf:params:scim:schemas:oracle:ids:extension:OCITags",
      "urn:ietf:params:scim:schemas:oracle:ids:extension:group:Group"
    ],
    "domainOcid":
      "ocid1.domain.oc1..aaaaaaaatzwhla6vhqwnvj6eiaoksjwtq2sxxwzls6zljr3ttcs5mj6kgk
      a",
    "compartmentOcid":
      "ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
      4a",
    "tenancyOcid":
      "ocid1.tenancy.oc1..aaaaaaa5oqwziy4bngiebry6letze4s2ap5x5pv4esurhc6pojwe4wx3
      4a",
    "urn:ietf:params:scim:schemas:oracle:ids:extension:dynamic:Group": {
      "membershipType": "static"
    }
  }
},
"startIndex": 1,
"itemsPerPage": 50
}

```

3.10 Delete a group

CNC Console uses Delete a group REST API to delete a group.

Type: DELETE

URI:

/admin/v1/Groups/{{groupid}}

Sample URI:

https://<oci-iam-domain-url>/admin/v1/Groups/{{groupid}}

Sample group:

```

{{
  "displayName": "ExampleGroup",
  .
  .
  .
  "id": "f4fc...6ad",
  "ocid": "ocid1.group.oc1...4fq",
  .
  .
  .
}}

```

Populate {{groupid}} as "id" of sample group

Example curl command:

```
curl --location --request DELETE 'https://<oci-iam-domain-url>/admin/v1/Groups/f4fc1efec8a1478fbe858cbbe95456ad' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsI....._LcCZYwDQJJTloj2PJ8y1Wj09l2Q'
```

Example of the Response Body

The following is the example of the response body:

204 No Content

3.11 Update group details

CNC Console uses Update group details REST API to update group details.

Type: PATCH

URI:

/admin/v1/Groups/{{groupid}}

Sample URI:

https://<oci-iam-domain-url>/admin/v1/Groups/{{groupid}}

Sample group:

```
{
  "displayName": "ExampleGroup",
  .
  .
  .
  "id": "f4fc...6ad",
  "ocid": "ocid1.group.oc1...4fq",
  .
  .
  .
}
```

Populate {{groupid}} as "id" of sample group

Example curl command:

```
curl --location --request PATCH 'https://<oci-iam-domain-url>/admin/v1/Groups/efac9f16bf5949318e36c32f6c4b4574' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsI....._LcCZYwDQJJTloj2PJ8y1Wj09l2Q' \
--header 'Content-Type: application/json' \
--data '{
```

```

"schemas": [
  "urn:ietf:params:scim:api:messages:2.0:PatchOp"
],
"Operations": [
  {
    "op": "replace",
    "path": "displayName",
    "value": "Employees from Example"
  }
]
}'

```

Example of the Response Body

The following is the example of the response body:

```

{
  "displayName": "Employees from Example",
  "idcsLastModifiedBy": {
    "type": "User",
    "value": "5427ea861e2c479895b8375b19c6827a",
    "display": "Test User",
    "ocid":
"ocid1.user.oc1..aaaaaaaaxszu5pi3iudjfa2tkkz3onobfnk3ctuqqyg3zsa7gx4sohikrboq"
  },
  "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
5427ea861e2c479895b8375b19c6827a"
},
  "idcsCreatedBy": {
    "type": "User",
    "ocid": "ocid1.user.oc1...krboq",
    "display": "Test User",
    "value": "5427ea861e2c479895b8375b19c6827a",
    "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/
5427ea861e2c479895b8375b19c6827a"
},
  "urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group": {
    "description": "Example Description"
  },
  "meta": {
    "version": "7a72fc34066347a9aafccf7258614c4d",
    "created": "2024-01-24T08:50:30.209Z",
    "lastModified": "2024-01-24T09:47:34.308Z",
    "resourceType": "Group",
    "location": "https://<oci-iam-domain-url>:443/admin/v1/Groups/
efac9f16bf5949318e36c32f6c4b4574"
  },
  "id": "efac9f16bf5949318e36c32f6c4b4574",
  "ocid": "ocid1.group.oc1...woq",
  "externalId": "123456",
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group",
    "urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group"
  ],
  "domainOcid": "ocid1.domain.oc1...j6kgka",

```

```

    "compartmentOcid": "ocid1.tenancy.oc1...xe34a",
    "tenancyOcid": "ocid1.tenancy.oc1...wxe34a"
  }

```

3.12 Assign group to a user

CNC Console uses Assign group to a user REST API to assign a group to a user.

Type: PATCH

URI:

/admin/v1/Groups/{{groupid}}

Sample URI:

https://<oci-iam-domain-url>/admin/v1/Groups/{{groupid}}

Sample group:

```

{{
  "displayName": "ExampleGroup",
  .
  .
  .
  "id": "f4fc...6ad",
  "ocid": "ocid1.group.oc1...4fq",
  .
  .
  .
}}

```

Populate {{groupid}} as "id" of sample group

Example curl command:

```

curl --location --request PATCH 'https://<oci-iam-domain-url>/admin/v1/Groups/efac9f16bf5949318e36c32f6c4b4574' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsI....._LcCZYWDQJJTloj2PJ8y1Wj0912Q' \
--data '{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:PatchOp"
  ],
  "Operations": [
    {
      "op": "add",
      "path": "members",
      "value": [
        {
          "value": "<userid>",
          "type": "User"
        }
      ]
    }
  ]
}'

```

```
    ]
  }
]
}'
```

Sample user format:

```
{
  "id": 12...x3,
  ...
  "ocid": "ocid1...op",
  "userName": "testuser"
  .
  .
  .
}
```

Populate <userid> as "id" of Sample User

Example of the Response Body

The following is the example of the response body:

```
{
  "displayName": "Employees from Example",
  "idcsLastModifiedBy": {
    "type": "User",
    "value": "5427ea861e2c479895b8375b19c6827a",
    "display": "Test User",
    "ocid": "ocid1.user.oc1...krboq",
    "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/5427ea861e2c479895b8375b19c6827a"
  },
  "idcsCreatedBy": {
    "type": "User",
    "ocid": "ocid1.user.oc1...krboq",
    "display": "Test User",
    "value": "5427ea861e2c479895b8375b19c6827a",
    "$ref": "https://<oci-iam-domain-url>:443/admin/v1/Users/5427ea861e2c479895b8375b19c6827a"
  },
  "urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group": {
    "description": "Example Description"
  },
  "meta": {
    "version": "7a72fc34066347a9aafccf7258614c4d",
    "created": "2024-01-24T08:50:30.209Z",
    "lastModified": "2024-01-24T09:47:34.308Z",
    "resourceType": "Group",
    "location": "https://<oci-iam-domain-url>:443/admin/v1/Groups/efac9f16bf5949318e36c32f6c4b4574"
  },
  "id": "efac9f16bf5949318e36c32f6c4b4574",
  "ocid": "ocid1.group.oc1...oewoq",
  "externalId": "123456",
```

```

    "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:Group",
      "urn:ietf:params:scim:schemas:oracle:ids:extension:group:Group"
    ],
    "domainOcid": "ocid1.domain.oc1...j6kgka",
    "compartmentOcid": "ocid1.tenancy.oc1...wxe34a",
    "tenancyOcid": "ocid1.tenancy.oc1...4wxe34a"
  }

```

3.13 Assign multiple groups to a user

CNC Console uses Assign multiple groups to a user REST API to assign multiple groups to a user.

Type: POST

URI:

/admin/v1/Bulk

/admin/v1/Groups/{{groupid}}

Sample URI:

https://<oci-iam-domain-url>/admin/v1/Bulk

Example curl command:

```

curl --location --request PATCH 'https://<oci-iam-domain-url>/admin/v1/Groups/efac9f16bf5949318e36c32f6c4b4574' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsI....._LcCZYwDQJJTloj2PJ8y1Wj09l2Q' \
--data '{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:PatchOp"
  ],
  "Operations": [
    {
      "op": "add",
      "path": "members",
      "value": [
        {
          "value": "<userid>",
          "type": "User"
        }
      ]
    }
  ]
}'

```

Example of the Response Body

The following is the example of the response body:

```
curl --location 'https://<oci-iam-domain-url>/admin/v1/Bulk' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsI....._LcCZYwDQJJTloj2PJ8y1Wj09l2Q' \
--data '{"schemas":["urn:ietf:params:scim:api:messages:2.0:BulkRequest"],
"Operations":
[
{
"method":"PATCH","path":"/Groups/<groupid>",
"data":{"schemas":["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
"Operations":[{"op":"add","path":"members","value":
[{"value":"<userid>","type":"User"}]}]}
},
{
"method":"PATCH","path":"/Groups/<groupid>",
"data":{"schemas":["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
"Operations":[{"op":"add","path":"members","value":
[{"value":"<userid>","type":"User"}]}]}
}
]
}'
```

Populate different groups with <groupid>, to assign multiple groups to user, <userid>

Sample user format:

```
{
  "id": 12...x3,
  ...
  "ocid": "ocid1...op",
  "userName": "testuser"
  .
  .
  .
}
```

Populate <userid> as "id" field of sample User

Sample group format:

```
{
  "displayName": "ExampleGroup",
  .
  .
  .
  "id": "f4fc...6ad",
  "ocid": "ocid1.group.oc1...4fq",
  .
  .
  .
}
```

Populate <groupid> as "id" of sample group