

# INSTITUTE OF COMPUTER TECHNOLOGY

## B-TECH COMPUTER SCIENCE ENGINEERING 2025-26

### SUBJECT: IDENTITY ACCESS MANAGEMENT

NAME: Rahul Prajapati

ENRLL NO: 23162171020

BRANCH: CYBER SECURITY

BATCH: 52

### Lab 5: Realm Management and Client Management

#### STEP\_1:Start Container

- Open docker desktop.
- Start container.
- Open browser and go to: <http://localhost.com:8080>

Name	Display name
master (Current realm)	Keycloak

#### STEP\_2:Create Realm

- Login as admin.
- Go to manage realm → create realm.
- Enter realm name and Save.

**Create realm**

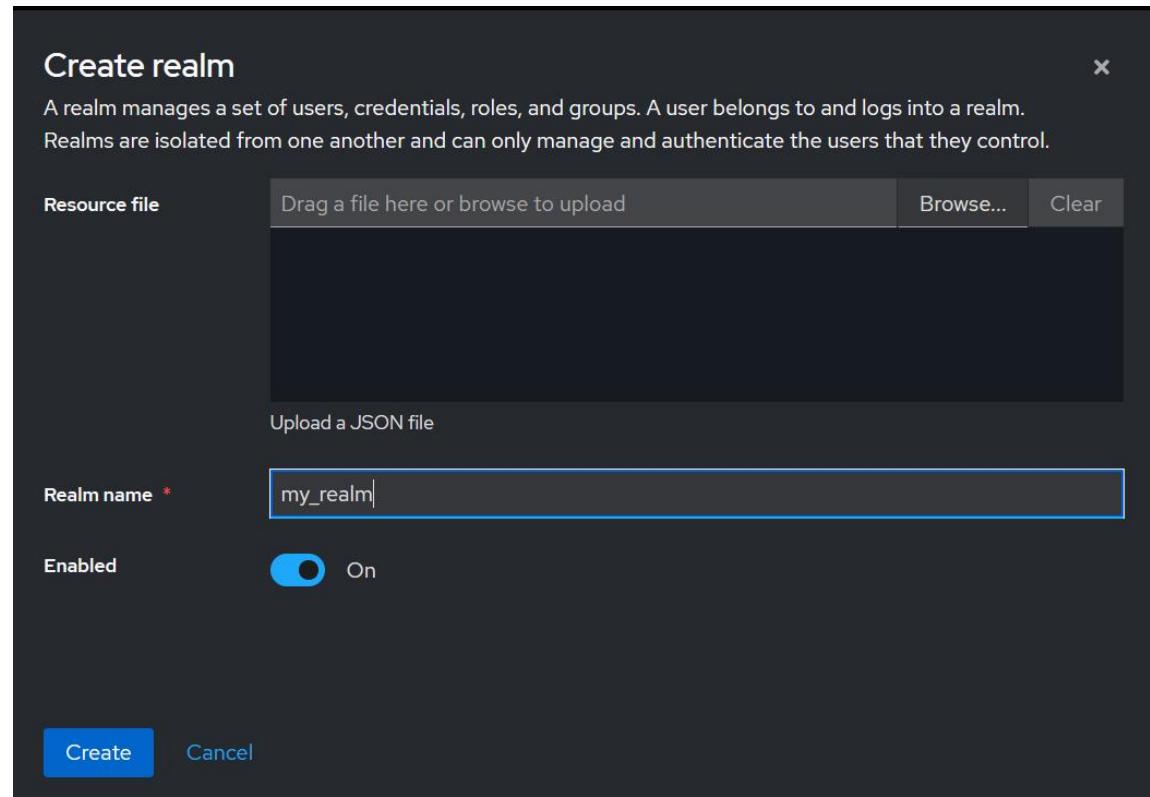
A realm manages a set of users, credentials, roles, and groups. A user belongs to and logs into a realm. Realms are isolated from one another and can only manage and authenticate the users that they control.

**Resource file** Drag a file here or browse to upload

Upload a JSON file

**Realm name \*** my\_realm

**Enabled**  On



## STEP\_3:Create Client

- Navigate to clients → create client.
- Configure authentication, redirect URLs and Save.

1 General settings

2 Capability config

3 Login settings

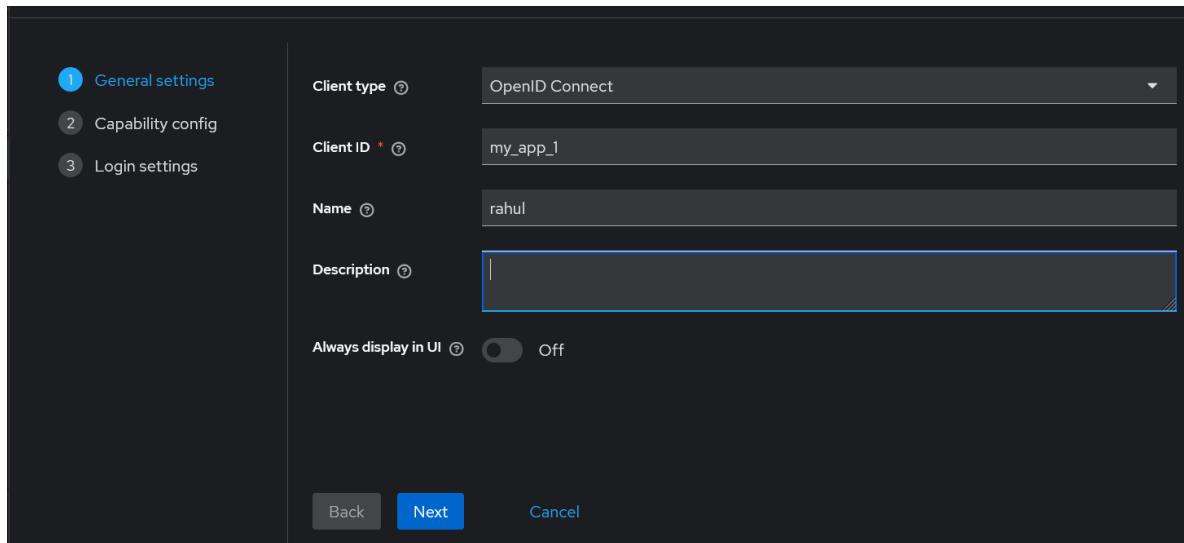
**Client type** OpenID Connect

**Client ID \*** my\_app\_1

**Name** rahul

**Description**

**Always display in UI**  Off



## STEP\_4:Create Client Role

- Go to the create client → roles
- Create a role with name & description

Clients > Client details > Role details

## manager

	Details	Associated roles	Attributes	Users in role	Admin events
Role name	manager				
Description	developer team's manager				

**Save** **Cancel**

## Step\_5:Assign Client Role

- Go to Users → Role Mappings.
- Assign the client role to user.

Assign Client roles to rahulbhai

<input type="checkbox"/> delete-account	account	role_delete-account
<input type="checkbox"/> manage-account	account	role_manage-account
<input type="checkbox"/> manage-account-links	account	role_manage-account-links
<input type="checkbox"/> manage-consent	account	role_manage-consent
<input type="checkbox"/> view-applications	account	role_view-applications
<input type="checkbox"/> view-consent	account	role_view-consent
<input type="checkbox"/> view-groups	account	role_view-groups
<input type="checkbox"/> view-profile	account	role_view-profile
<input type="checkbox"/> read-token	broker	role_read-token
<input checked="" type="checkbox"/> manager	my_app_1	developer team's manager

1 - 10 ▾ < >

**Assign** **Cancel**

## STEP\_6:Verify Assigned Roles

- ➔ Go to clients → Roles →Users in role.
- ➔ Check assigned user.

Role name	Composite	Description	⋮
manager	False	developer team's manager	⋮

## Explanation of Commands/Steps

**Realm** – A space where you can manage users, roles, and applications.

**Client** – An app or service that uses Keycloak for login and access.

**Client Role** – A role that belongs to a specific client (app/service).

**Role Mapping** – Connecting a user to a role so they have certain permissions.

## Conclusion

In this lab, we did the following:

- Launched a Keycloak container using Docker.
- Created a Realm to manage user access and permissions.
- Set up a Client (app/service) to handle authentication.
- Created and assigned a role to the client.
- Checked that users and roles were linked properly.