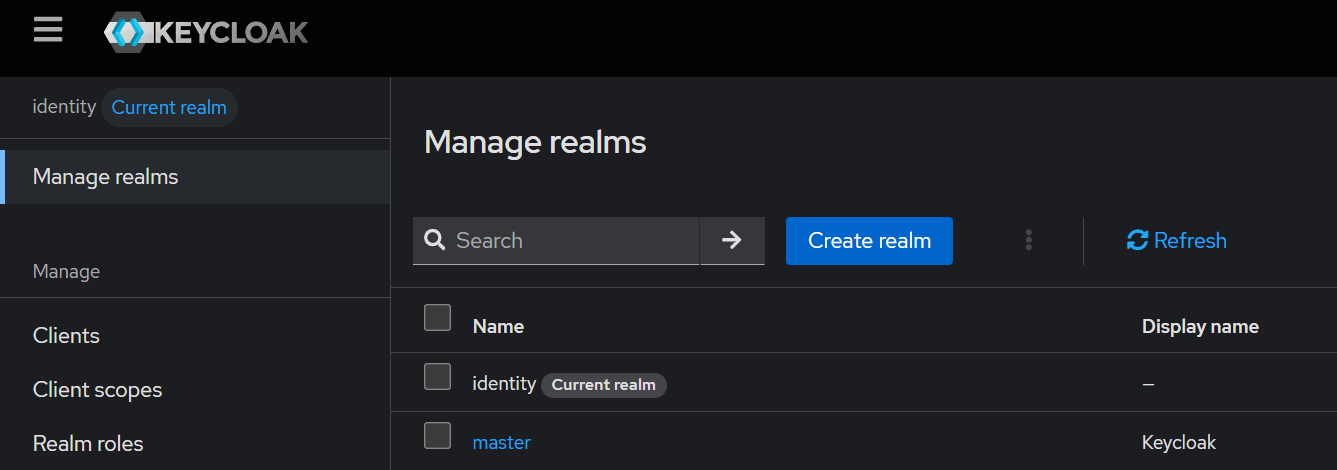
## Design Approach

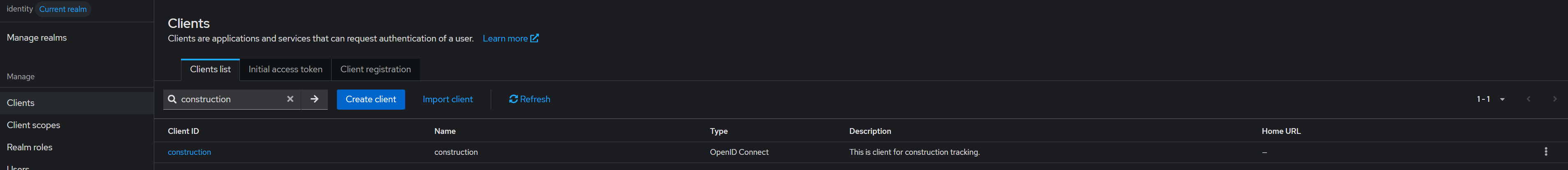
* Cross-user multi-tenant application (i.e. users can be shared across tenants if needed)
* Global admin user should be able to manage all tenant.
* Local admin should be able to manage single tenant.
* user consumer of one or more tenant.

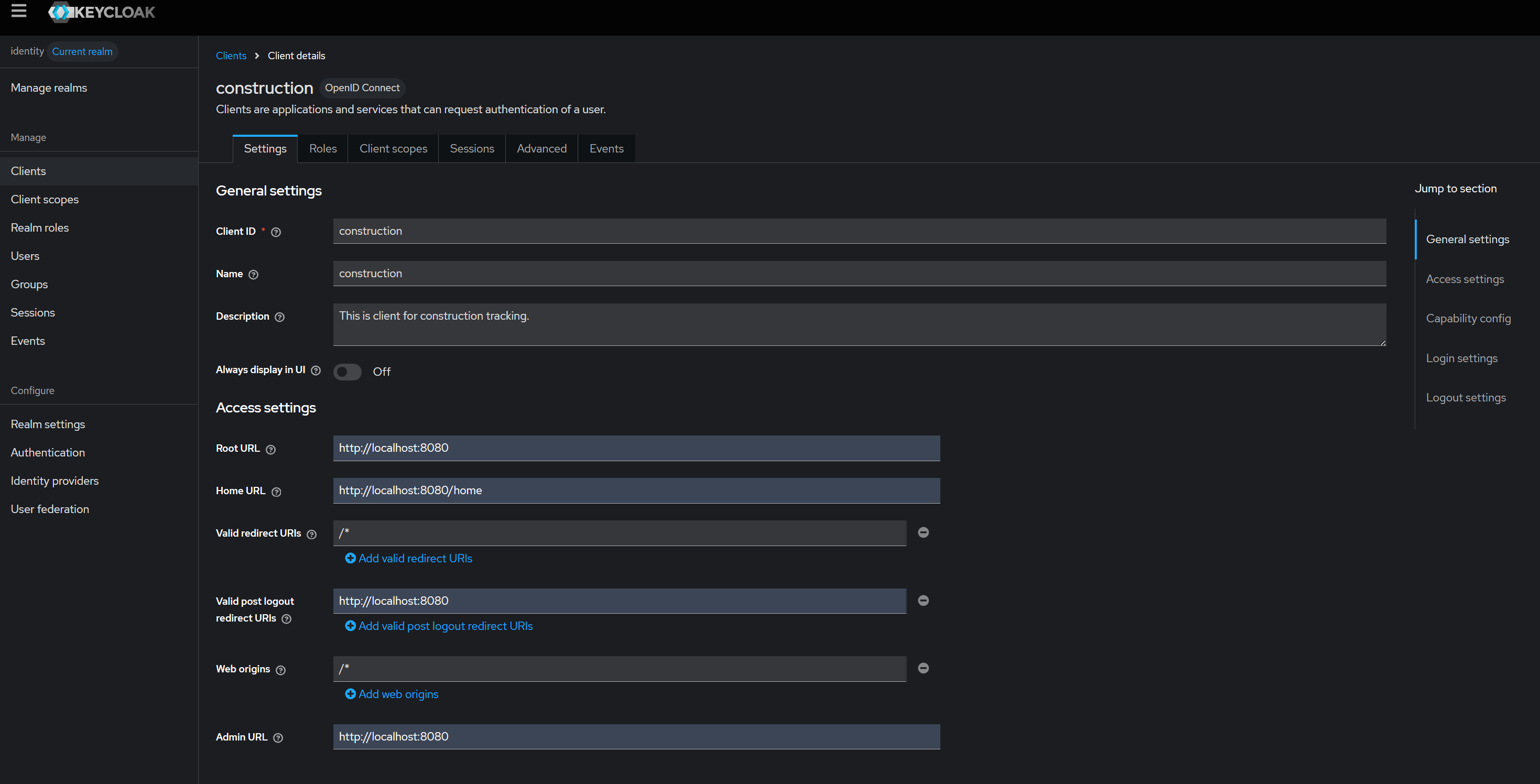
|  |  |  |  |
| --- | --- | --- | --- |
| **KeyCloak Identity** | **Rational** | **Examples** | **values** |
| Realm | Single Realm for Central Identity |  | identity |
| Clients | One per Application i.e Each application is a Keycloak client. | client\_app\_1, client\_app\_2 | construction |
| Groups | Model Tenants | tenant1, tenant2 | dlf, goderj |
| Child group | Projects of each tenants |  |  |
| Child-Child group | Departments etc for each tenant | Finance, construction etc |  |
| Roles | Model Role Hierarchy | Global roles: app-admin, local-admin, user  Client roles: client\_app\_1\_user, client\_app\_1\_admin, etc. |  |

* creating realm



* Creating client





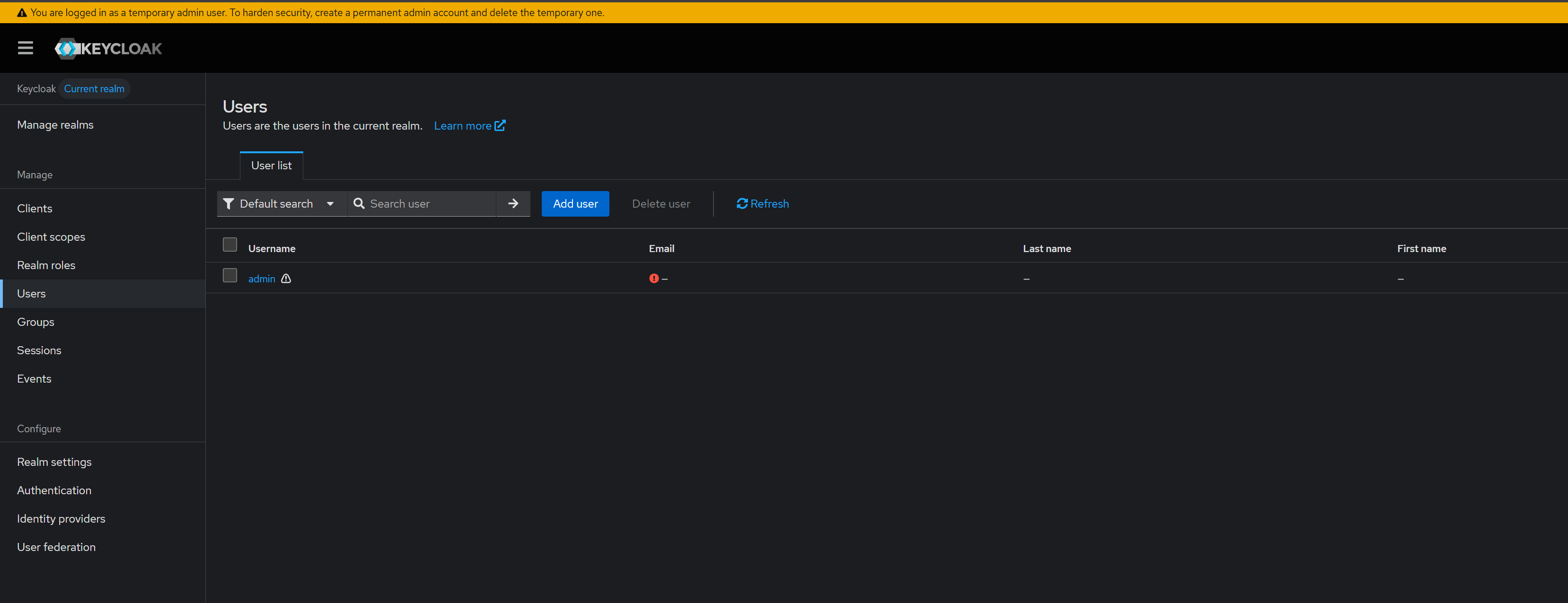
## Setup in dev env

1. Install podman desktop
2. Download and install keycloak image in podman

cmd: ``` podman run -d --name Identity\_keycloak -e KC\_BOOTSTRAP\_ADMIN\_USERNAME=admin -e KC\_BOOTSTRAP\_ADMIN\_PASSWORD=admin -p 127.0.0.1:53551:8080 quay.io/keycloak/keycloak:26.4.0 start-dev ```

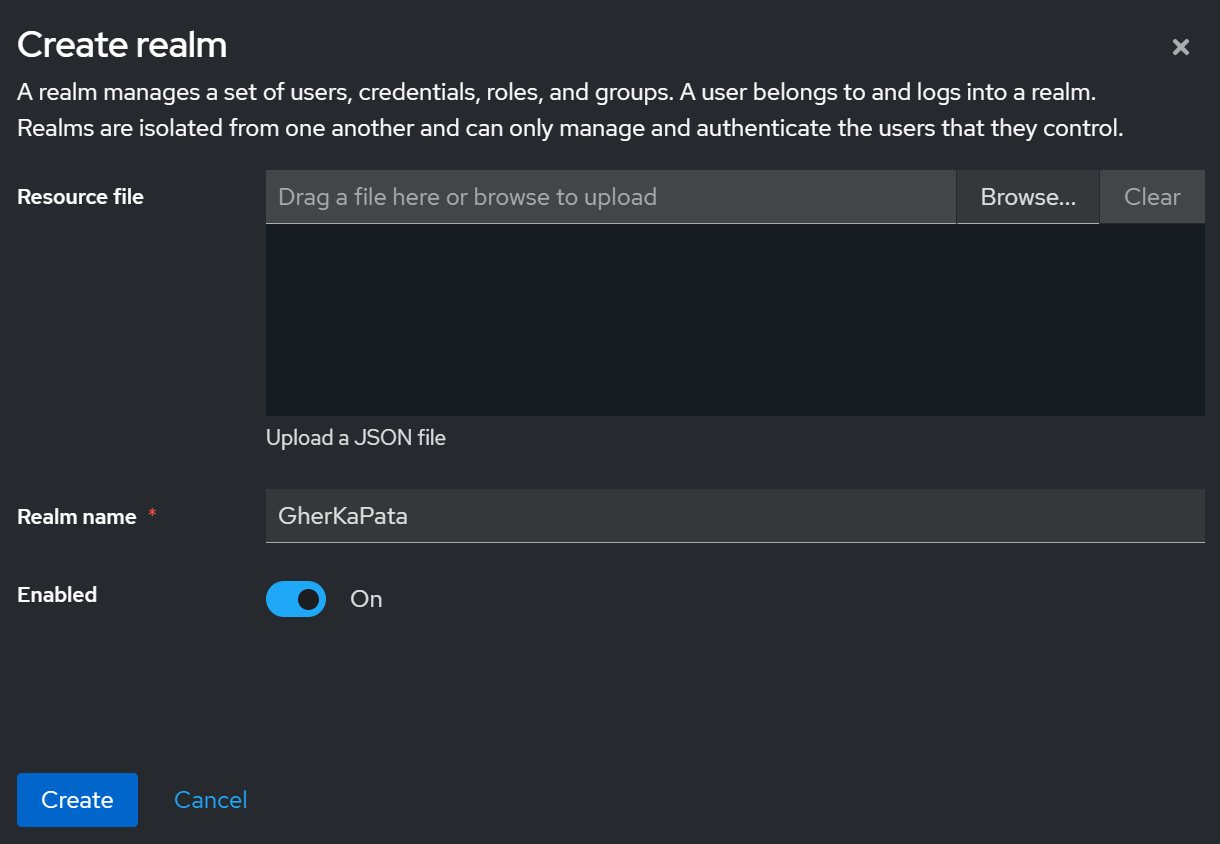
|  |  |
| --- | --- |
| **Part** | **Meaning** |
| podman run | Runs a new container using Podman. |
| -d | Detached mode — runs the container in the background. |
| --name boldmclaren\_new | Names the container boldmclaren\_new. |
| -e KC\_BOOTSTRAP\_ADMIN\_USERNAME=admin | Sets the Keycloak admin username to admin. |
| -e KC\_BOOTSTRAP\_ADMIN\_PASSWORD=admin | Sets the Keycloak admin password to admin. |
| -p 127.0.0.1:53551:8080 | Forwards port 8080 in the container to localhost:53551 on the host — only accessible locally. |
| quay.io/keycloak/keycloak:26.4.0 | The image to use (Keycloak version 26.4.0, pulled from Quay container registry). |
| start-dev | Tells Keycloak to start in **development mode** (no HTTPS, faster boot, etc.). |

1. Configuring keycloak
   1. Create permanent admin user, to avoid error: “You are logged in as a temporary admin user. To harden security, create a permanent admin account and delete the temporary one.”

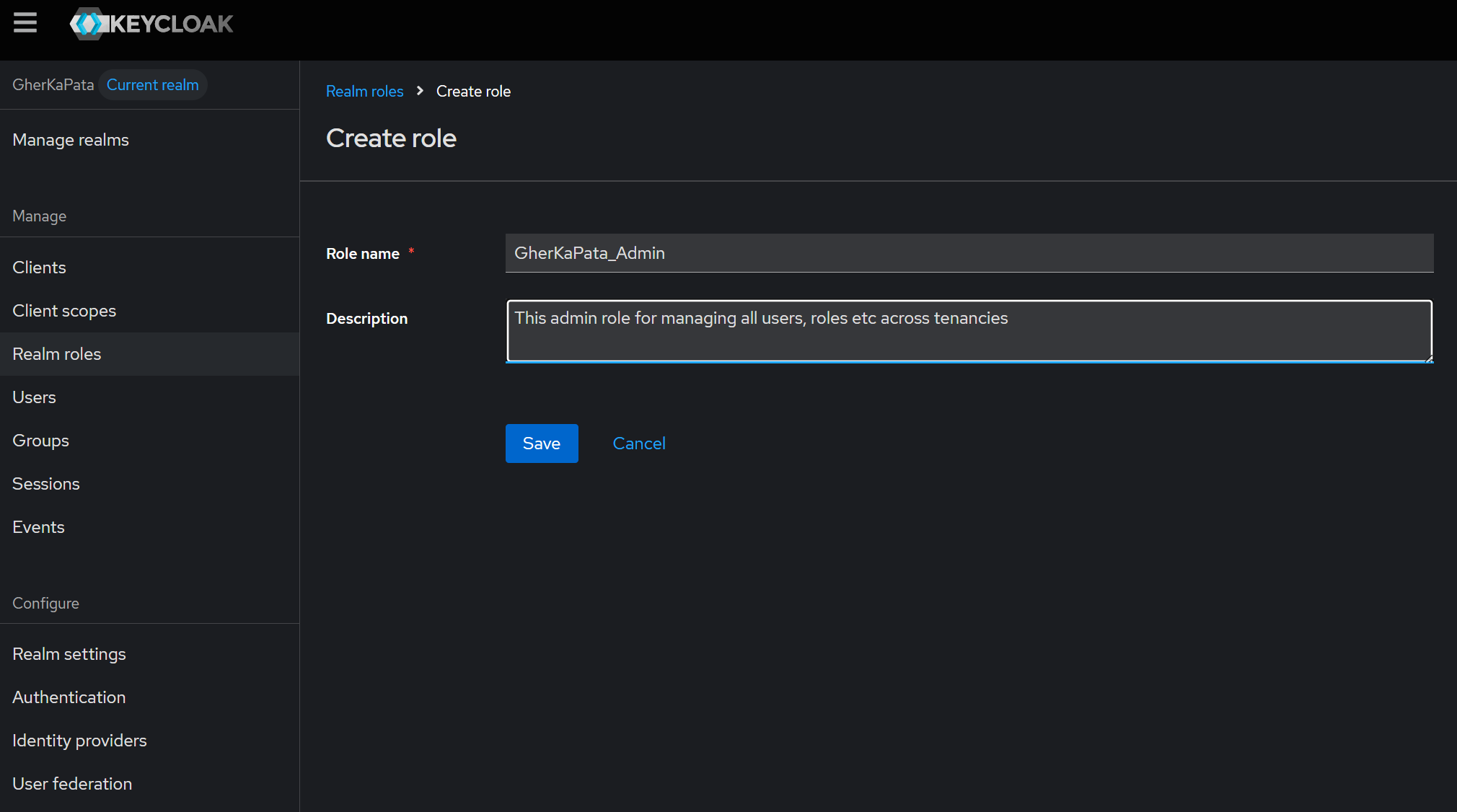


For this create a new user add all relam roles including admin and other roles. Login with that user and remove old admin user.

* 1. Creating realm

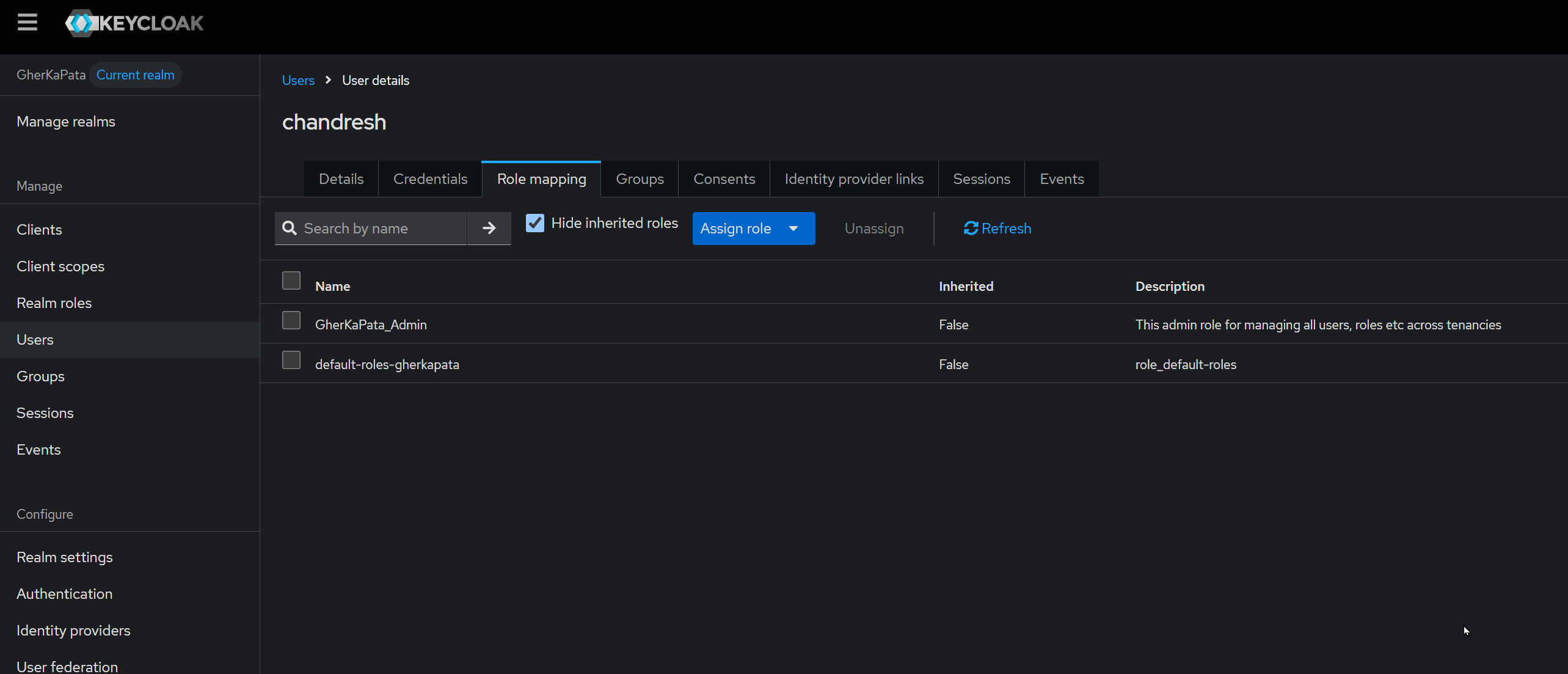


* 1. Created role for global admin

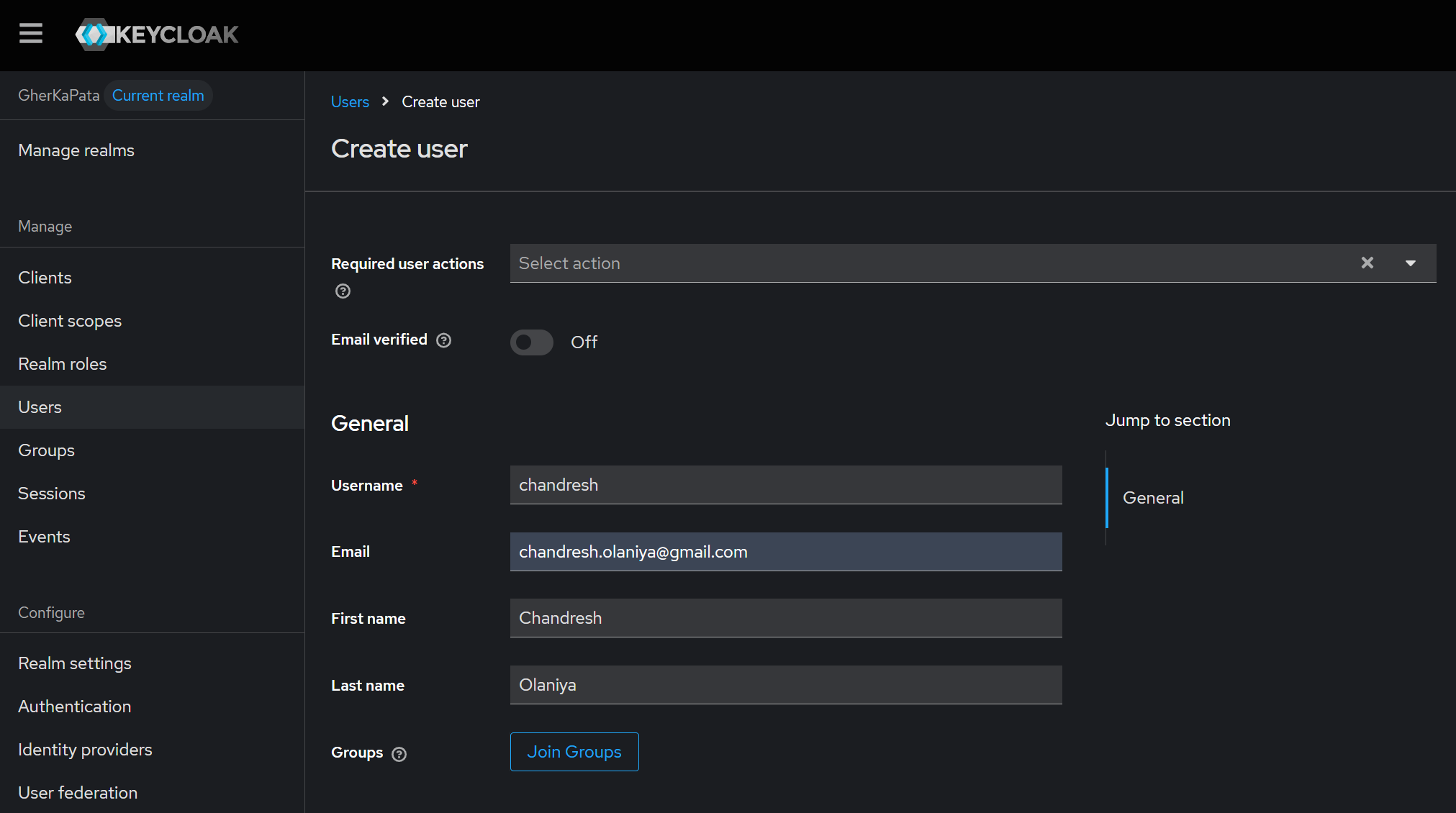


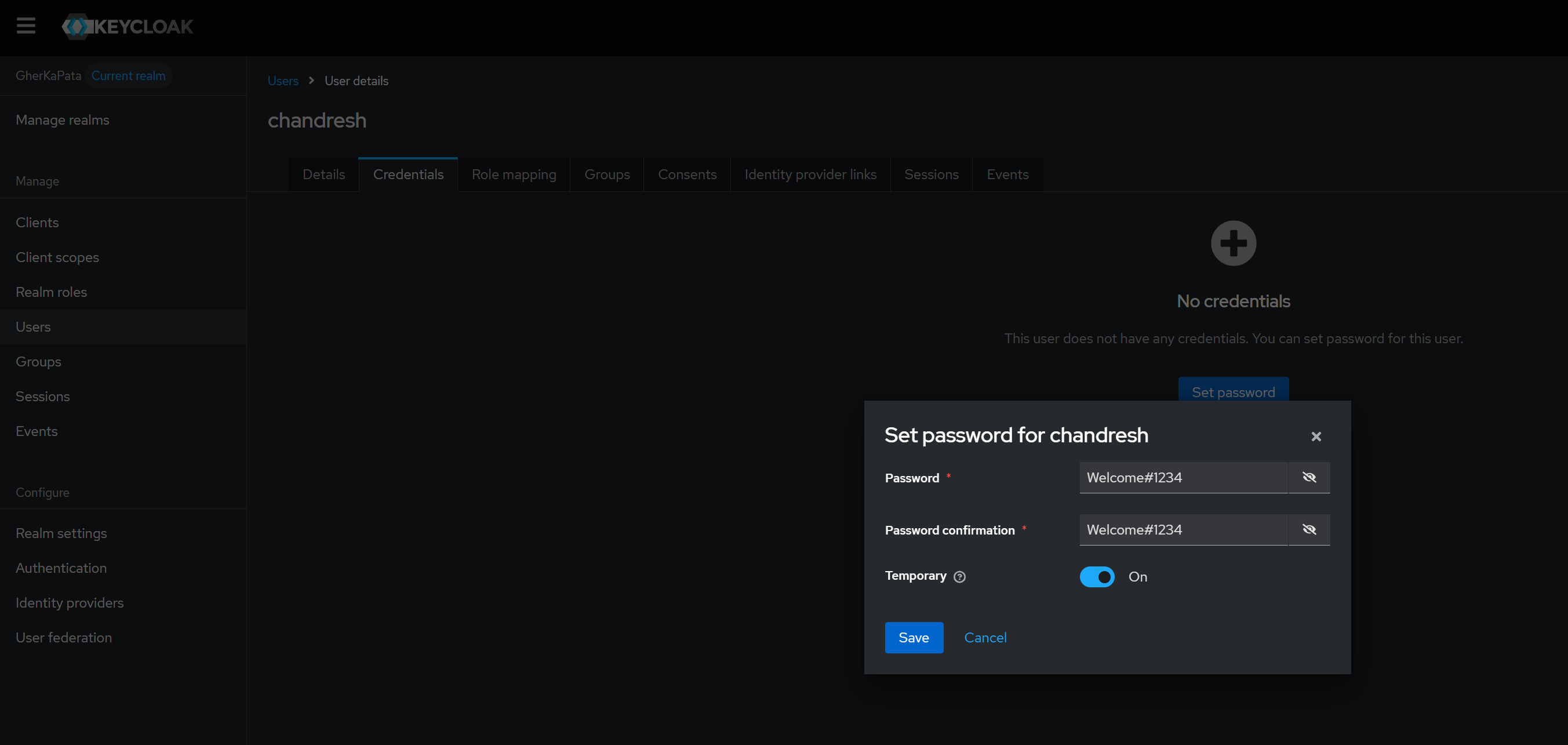
Currently this user have all client and realm roles assigned to it.

* 1. Added global\_admin role to this user

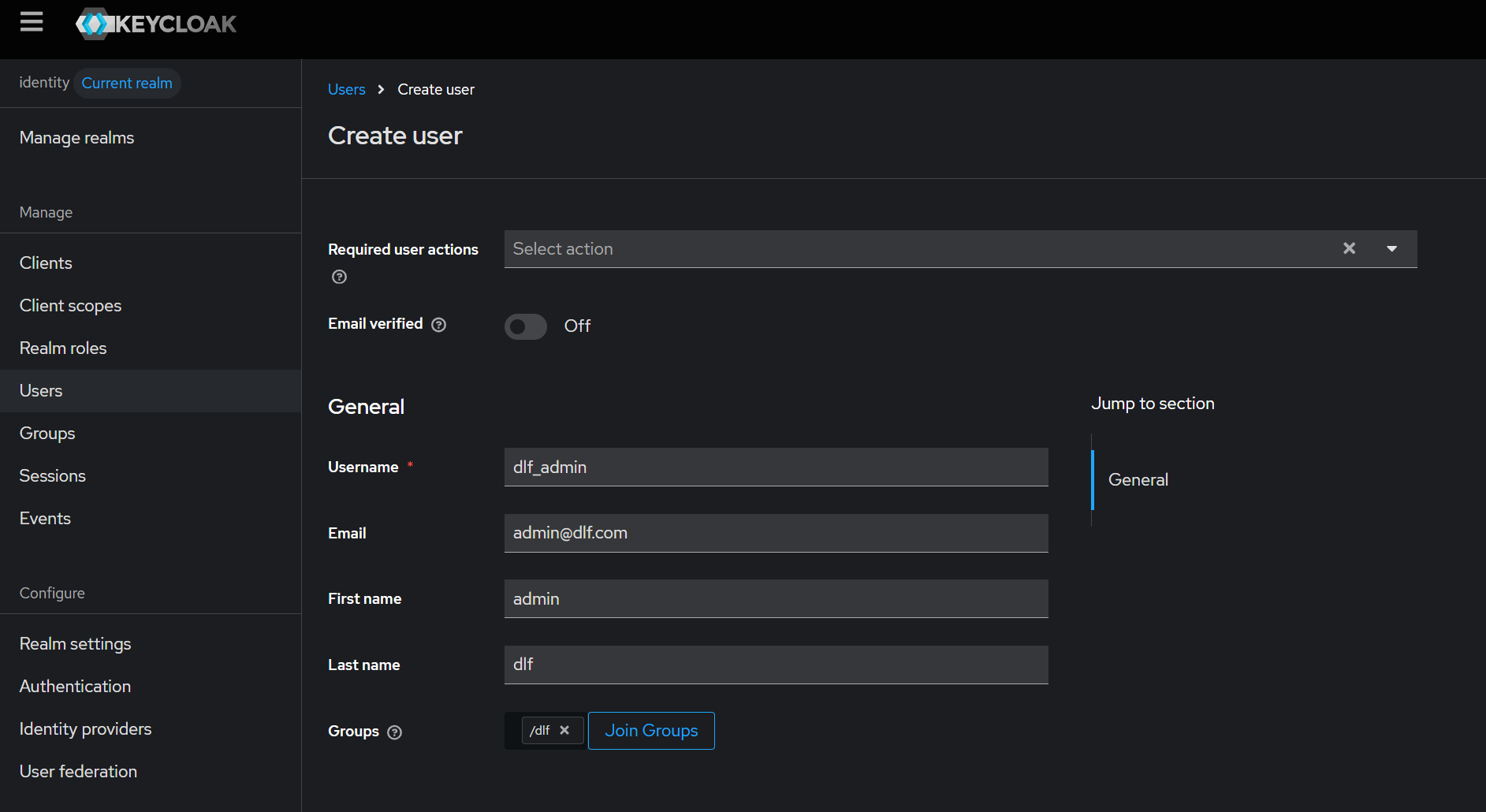


* 1. Creating user for the global admin role





* 1. Creating users
     1. builder\_admin



* + 1. builder\_user

1. Configuration in VueJS
   1. I am using '@josempgon/vue-keycloak'
   2. Created keycloak.config.ts in './components/auth/keycloak.config'

// This is the configuration file for Keycloak

// Optional: You can install the 'keycloak-js' types: npm install --save-dev @types/keycloak-js

import type Keycloak from 'keycloak-js'

// Recommended way to get the base URL dynamically

const appBaseUrl: string = window.location.origin

const keycloakServerBaseUrl = 'http://127.0.0.1:53551'

// ⚠️ Explicitly define the type for the config object

const keycloakConfig: Keycloak.KeycloakConfig = {

// Note: 'http://127.0.0.1:53551' looks like a temporary Vite/Dev server address.

// Ensure this is your actual Keycloak server URL/port.

url: `${keycloakServerBaseUrl}`, // Use your confirmed server address + /auth

realm: 'GherKaPata',

clientId: 'VueApp'

}

const keycloakInitOptions = {

// 'login-required' forces a login on app load. 'check-sso' attempts silent login.

// commenting this for check 404 error

onLoad: 'login-required',

// 🔑 Crucial Fix: Set this to false to stop the adapter from attempting the

// third-party cookie iframe check (3p-cookies/step1.html).

// This resolves the 404 and the subsequent Timeout error.

checkLoginIframe: false,

redirectUri: `${appBaseUrl}/home`, // ✅ Redirect to /home (not /login)

// This is optional but highly recommended for refreshing tokens silently in the background

silentCheckSsoRedirectUri: `${appBaseUrl}/silent-check-sso.html`

}

export { keycloakConfig, keycloakInitOptions }

## Important URL’s

|  |  |
| --- | --- |
|  |  |
| Keycloak Account Console | http://localhost:53551/realms/GherKaPata/account |
|  |  |
|  |  |
|  |  |