Collaborative Data Hub Software - Maintenance and Evolution Services - Ready for Digital Twin Earth

Keycloak Installation and Configuration Manual

Role/Title	Name	Signature	Date
Authors	Giuseppe Presta, Giulia Carosi		28/05/2024
Verified/Approved	Barbara Borgia Matteo Cortese		28/05/2024

Change register

Version /Rev.	Date	Section	Description
1.0	04/02/2022	All	First release of the document
1.1	11/04/2022	1.3	Updated GDPR TN issue to the latest version
		1.4	Updated DHuS Administration Manual issue to the latest version
		2	Typo correction Removed statement that the theme is valid only in case of one client configured
		3.2	Typo correction
		4.1	Removed link to download Keycloak theme Typo correction Updated parameters to be set due to theme update
		4.2.1.2	Typo correction
		4.2.2	Typo correction Added AUTHED as DHuS User Roles in Table 6 Updated Figure 8 due to AUTHED added Role Added Steps to avoid receiving User Roles from other clients
		4.2.4	Typo correction
		4.2.7	Updated Figure 21 due to theme update
		4.2.8	Replaced Figure 25 with the correct one
		5.1	Removed link to download .jar file
		5.2	Removed link to download script for User migration
		5.2.3.1	Typo correction
		5.2.4.1	Removed link to download script to execute queries on DHuS embedded DB
2.0	16/05/2022	3, 4	Document updated to insert instruction of how to configure Keycloak to interface with Transformation Framework component.
2.1	11/07/2022	4.1	Adding User Role Scope Mappings setting
2.2	22/03/2023	5	Document updated to insert instruction of how to configure Keycloak to interface with GAEL Store Service component.
2.3	06/10/2023	1.4	Reference Document versions updated

2.4	28/05/2024	1.4	Updated Transformation Framework and GSS documents to the latest versions
		2, 2.2, 2.3	Updated Keycloak version
		5.1.2	Added procedure to SSO Authentication for the GSS Software
		5.1.3	Updated procedure to assign roles to GSS Users

Table of Contents

Table Index

Figure Index

1. Introduction

1.1 Purpose and Scope

The purpose of this document is to describe how to install and configure Keycloak as Identity Access Management System provider for the DHS components in the frame of the "Collaborative Data Hub Software Maintenance and Evolution Services for Digital Twin Earth".

1.2 Acronyms and Abbreviations

Acronym	Description
DHuS	Data Hub System
SW	Software
RAM	Random Access Memory
GDPR	General Data Protection Regulation
SMTP	Simple Mail Transfer Protocol
MD5	Message Digest algorithm 5
VM	Virtual Machine
DB	Database
GUI	Graphical User Interface
GSS	GAEL Store Service

1.3 Applicable Documents

The following table lists the documents with a direct bearing on the content of this document.

Reference	Document Title	Reference	Issue
AD-1	https://www.keycloak.org/		
AD-2	GDPR Implementation	GAEL-P286-TCN-031	1.9

Table 1: Applicable Documents

1.4 Reference Documents

The following reference documents contain information supporting this document.

Reference	Document Title	Reference	Issue
RD 1	GDPR Compliancy for User Account Operational Scenarios	COPE-SERCO-TN-21-1097	1.1
RD 2	DHuS Administration Manual	SPA-COPE-DHUS-UM-001	2.11
RD 3	Collaborative Data Hub Software - Transformation Framework Installation Configuration Manual	COPE-SERCO-TN-21-1218	4.1
RD 4	Collaborative Data Hub Software GSS Administration Manual	GAEL-P311-GSS-CDH-Administration Manual	1.6.5
RD 5	Collaborative Data Hub Software GSS Software Design Document	GAEL_P311 – GSS-CDH-SDD	1.7.3

Table 2: Reference Documents

2. Keycloak Installation

This Chapter provides guidelines to install Keycloak in different modes according to Operational needs:

- Installation in legacy mode with an embedded H2 Database
- Installation in legacy mode with a Postgres externalized Database

All the installation modes refer to Keycloak 18.0.2 version which is the one currently supported by the DHuS SW.

Please note that the steps described in next Sections allow to install Keycloak in a standalone operating mode, which means that only one Keycloak instance is running.

If you want to deploy Keycloak in a standalone clustered operation mode, please refer to Keycloak Website [AD-1] for more details.

Please refer to the official Website [AD-1] also for Keycloak installation via Docker.

2.1 Pre-requirements

The following preconditions shall be satisfied to proper install Keycloak:

- Java 8 JRE or Java 11 JRE installed
- Zip or gzip and tar modules installed
- At least 512M of RAM available
- At least 1G of disk space available

2.2 Manual legacy – Embedded H2 DB

By default, Keycloak comes with its own embedded Java-based relational database called H2. This is the default database that Keycloak uses to persist data and it allows to run instantly the authentication server.

The Table below summarizes the steps to perform for installing Keycloak in standalone mode with its own embedded Database.

Step ID	Actions
1	Download from the official website the Keycloak v18.0.2 in .tar.gz format: https://www.keycloak.org/archive/downloads-18.0.2.html
2	Upload the downloaded package in your installation folder and decompress it: `> tar -xvzf keycloak-18.0.2.tar.gz`
3	Access the /standalone/configuration folder and modify the standalone.xml file to configure the address of the VM where Keycloak is installed. Below the sections in bold to modify: • Replace with the FrontEnd URL towards which your Keycloak instance is exposed: `<property name="frontendUrl" value="`**`${keycloak.frontendUrl:}`**`"/>`

	• Replace with the IP address of the VM where your Keycloak instance is installed: <inet-address value="${jboss.bind.address.management:**127.0.0.1**}"/> <inet-address value="${jboss.bind.address:**127.0.0.1**}"/> • If not 8080, replace with the PORT towards which your Keycloak instance is exposed: <socket-binding name="http" port="${jboss.http.port:**8080**}"/>
4	*[Optional – Only if your Keycloak instance is exposed towards a Proxy]* Add in the standalone.xml configuration file the followings lines marked in bold: • If your proxy is forwarding requests via the HTTP protocol, configure Keycloak to pull the client's IP address from the X-Forwarded-For header rather than from the network packet: <http-listener name="default" socket-binding="http" redirect-socket="proxy-https" enable-http2="true" **proxy-address-forwarding="true"** /> • Configure the HTTPS port traffic towards which traffic is redirected to: **<socket-binding name="proxy-https" port="443"/>**
5	Start Keycloak with the following command: > nohup ./bin/standalone.sh &> /dev/null &

Table 3: Installation Keycloak embedded DB

2.3 Manual legacy – Externalized PostgreSQL DB

Keycloak can be configured in order to support an externalized Database where data are stored, instead of the embedded one.

In this Section, how to setup Keycloak with an externalized PostgreSQL Database is described. If you want to install Keycloak with a different Database, please refer to the official WebSite [AD-1].

In addition to the listed preconditions in Section 2.1, the following shall be satisfied:

- PostgreSQL (> 10.14) installed

The Table below summarizes the steps to perform for installing Keycloak in standalone mode with an externalized PostgreSQL Database.

Step ID	Actions
1	Access your PostgreSQL server using root credentials and execute the following commands: • Create a new database: CREATE DATABASE keycloak; • Create a new user: CREATE USER <user> WITH ENCRYPTED PASSWORD '<password>'; • Grant all privileges to the created user: GRANT ALL PRIVILEGES ON DATABASE keycloak TO <user>; • Make the just created User owner of the Datatbase: ALTER DATABASE keycloak OWNER TO <user>;
2	Download the latest PostgreSQL JDBC driver from the following website:

	https://jdbc.postgresql.org/download.html
3	Download from the official website the Keycloak v18.0.2 in .tar.gz format: https://www.keycloak.org/archive/downloads-18.0.2.html
4	Upload the downloaded package in your installation folder and decompress it: `> tar -xvzf keycloak-18.0.2.tar.gz`
5	Inside the folder /modules/system/layers/keycloak/org, create a new folder postgresql/ with a folder main/ inside. The final path should be: `/modules/system/layers/keycloak/org/postgresql/main`
6	Upload the downloaded PostgreSQL driver .jar in the main/ folder.

7	In the main/ folder, create a file module.xml with the following content:
	
	`<?xml version="1.0" ?>`
	`<module xmlns="urn:jboss:module:1.3" name="org.postgresql">`
	
	`<resources>`
	`<resource-root path="`**`postgresql-42.2.19.jar`**`"/>`
	`</resources>`
	
	`<dependencies>`
	`<module name="javax.api"/>`
	`<module name="javax.transaction.api"/>`
	`</dependencies>`
	`</module>`
	
	Please substitute the bold .jar with the correct version downloaded at Step 2.

8	Access the /standalone/configuration folder and modify the standalone.xml file to configure the address of the VM where Keycloak is installed.
	
	Below the sections in bold to modify:
	• Replace with the FrontEnd URL towards which your Keycloak instance is exposed:
	`<property name="frontendUrl" value="`**`${keycloak.frontendUrl:}`**`"/>`
	• Replace with the IP address of the VM where your Keycloak instance is installed:
	`<inet-address value="${jboss.bind.address.management:`**`127.0.0.1`**`}"/>`
	`<inet-address value="${jboss.bind.address:`**`127.0.0.1`**`}"/>`
	• If not 8080, replace with the PORT towards which your Keycloak instance is exposed:
	`<socket-binding name="http" port="${jboss.http.port:`**`8080`**`}"/>`

9	*[Optional – Only if your Keycloak instance is exposed towards a Proxy]* Add in the standalone.xml configuration file the followings lines marked in bold:

	• If your proxy is forwarding requests via the HTTP protocol, configure Keycloak to pull the client's IP address from the X-Forwarded-For header rather than from the network packet: <http-listener name="default" socket-binding="http" redirect-socket="proxy-https" enable-http2="true" **proxy-address-forwarding="true"** /> • Configure the HTTPS port traffic towards which traffic is redirected to: **<socket-binding name="proxy-https" port="443"/>**
10	Update the standalone.xml configuration file to configure PostgreSQL as driver for Keycloak. • In the "drivers" section, add the JDBC driver for PostgreSQL: <driver name="postgresql" module="org.postgresql"> <xa-datasource-class>org.postgresql.xa.PGXADataSource</xa-datasource-class> </driver> • In the section "datasources", comment out the "KeycloakDS Datasource" and create a new Datasource for PostgreSQL Database. The section should be like below at the end: <datasources> <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true" statistics-enabled="${wildfly.datasources.statistics-enabled:${wildfly.statistics-enabled:false}}"> <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url> <driver>h2</driver> <security> <user-name>sa</user-name> <password>sa</password> </security> </datasource> <!--Comment the existing KeycloakDS--> <!-- <datasource jndi-name="java:jboss/datasources/KeycloakDS" pool-name="KeycloakDS" enabled="true" use-java-context="true" statistics-enabled="${wildfly.datasources.statistics-enabled:${wildfly.statistics-enabled:false}}"> <connection-url>jdbc:h2:${jboss.server.data.dir}/keycloak;AUTO_SERVER=TRUE</connection-url> <driver>h2</driver> <security> <user-name>sa</user-name> <password>sa</password> </security> <pool> <max-pool-size>100</max-pool-size> </pool> </datasource>--> <!--Adding new datasource for PostgreSQL--> <datasource jndi-name="java:jboss/datasources/KeycloakDS" pool-name="KeycloakDS" enabled="true" use-java-context="true"> <connection-url>jdbc:postgresql://localhost/keycloak</connection-url> <driver>postgresql</driver> <pool> <max-pool-size>20</max-pool-size> </pool> <security> <user-name><user></user-name>/ <password><password></password>

	</security> </datasource> ... </datasources> where <user> and <password> are those selected at Step 1.
11	Start Keycloak with the following command: > nohup ./bin/standalone.sh &> /dev/null &

Table 4: Installation Keycloak externalized PostgresSQL DB

2.4 Log files

After installed Keycloak in legacy mode, the application can be monitored thanks to the log file server.log generated in the folder /standalone/log/:

```
> tail -f standalone/log/server.log
```

3. DHuS Software

3.1 Overview

Keycloak is an Open Source Identity and Access Management solution which mainly aims at applications and services.

To cope with GDPR requirements, it has been selected as User Management provider for DHuS software.

The main scope of this document is to describe all needed actions to make Keycloak compliant with User interfaces currently implemented in DHuS SW:

- Chapter 2 is dedicated to Keycloak installation. How to install the SW in legacy mode with an embedded or externalized database is described.
- Chapter 3.2 is focused on Keycloak configuration. In particular, details on how to set up the User Management provider to cope with User functionalities currently supported by DHuS are provided.
- Chapter 3.3 is a guide for the migration of Users registered on DHuS instance to Keycloak.

Before to proceed with Keycloak's setup, a reading of the whole document is strongly suggested.

To better follow the instructions reported in this document, a summary of the suggested flow actions to be performed is provided below:

1. Install Keycloak according to the Operational needs (Chapter 2).
    a. Before start Keycloak, add the provided theme folder (Section 3.2.1).
    b. Before start Keycloak, if you want to migrate Users already registered on DHuS instance to Keycloak, add the .jar module to enable MD5 as password encrypt mechanism (Section 3.3.1).
2. Create the Administrator User to access the Keycloak administration console (Section 3.2.2.1).
3. Configure Keycloak in order to properly interface DHuS instance by creating Realm and client (Section 3.2.2.2).

    Please note that instructions described in the related Section are linked to actions to be executed on DHuS side.
4. Setup the password policy on Keycloak (Section 3.2.2.3).
    a. If you want to migrate Users already registered on DHuS instance to Keycloak, add also the policy linked to the MD5 password encrypt mechanism (Section 3.3.1).
5. Define the STMP Service to allow email notification in case of forgot password (Section 3.2.2.4).
6. In order to replicate the current User interfaces available on DHuS, select the provided theme for the Login and Account Keycloak functionality (Section 3.2.2.5, 3.2.2.6, 3.2.2.7, 3.2.2.8).
    a. To activate the 'Forgot Password' and 'Self Registration' fluxes, execute additional setups (Section 3.2.2.7, 3.2.2.8).
7. If you want to migrate already registered Users on Keycloak:
    a. Backup the Realm previously created (Section 3.3.2.2)

b. Export Users info stored in DHuS DB (Section 3.3.2.4)

c. Migrate Users on Keycloak running the provided script (Section 3.3.2.3)

Following what stated above, a proper User management via Keycloak is guarantee.

Please note that the provided theme refers to the definition of a single Realm on Keycloak.

3.2 Keycloak Configuration

In the framework of the DHuS Service, the GDPR requirements give to Keycloak the role of User Management System, both for actions involving Standard User and those performed by Administrator.

This means that through Keycloak it shall be possible to execute all actions allowing a Standard User to properly manage him/her own account and an Administrator User to correctly manage Users accounts.

In particular, Keycloak shall ensure the possibility to perform the following actions:

- Login/Logout
- Self Registration
- Edit Profile
- Forgot Password

In order to cope with what DHuS SW actually offers to Users, a Keycloak customization is required.

The customization acts both on Keycloak theme definition and Keycloak console setup.

Next Sections describe how to configure Keycloak to be compliant with actual DHuS Users interfaces for all concerns User Management actions.

3.2.1 Keycloak Theme Configuration

Keycloak provides theme support for web pages and emails. This allows customizing the look and feel of End-User facing pages so they can be integrated with DHuS SW applications.

A couple of themes are pre-built in Keycloak and come bundled with the distribution inside the theme/ folder:

- *base*: it contains HTML templates and message bundles. All themes, including custom ones, generally inherit from it.
- *keycloak*: it contains images and stylesheets for beautifying pages. If a custom theme is not provided, this is the one used by default.

In order to customize Keycloak to be compliant with current DHuS User Management functionalities, a dhus_theme/ theme has been generated.

Please refer to the relevant DHuS release notification to download it.

The provided folder shall be unzipped and uploaded in the theme/ folder inside the Keycloak installation folder. Please refer to Section 3.2.2 for details on how the theme configurations appear in the Keycloak Console interface.

In case the provided theme has to be modified, it is possible to update it and see the applied changes without restarting the Server by modifying as follows the theme section in the standalone.xml configuration file:

```
<theme>
    <staticMaxAge>-1</staticMaxAge>
    <cacheThemes>false</cacheThemes>
      <cacheTemplates>false</cacheTemplates>
      ...
</theme>
```

Please note that a Server restart is needed after this change.

The following Table summarizes the main configurable parameters useful to adapt Keycloak Console interface according to your Operational needs.

File/Folder	Parameter/Image	Description
/theme/keycloak/admin/resources/img	keycloak-logo.png	This image is the logo shown in the Admin console.
/theme/keycloak/welcome/resources	keycloak-logo.png	These images are the logo shown in the Welcome page.
	logo.png	
/theme/base/admin/messages/admin-messages_en.properties	consoleTitle	It allows to configure the message that appears at the mouse over on the Web Page title.
/theme/dhus_theme/login/messages/messages_en.properties	hubName	It allows to configure the title of the Login page.
	termsAndConditionLink	It allows to define the link redirecting to page where Term and Conditions are described.
	logoutUrl	It reports the Keycloak address to perform the logout.
	accountRedirectUrl	It reports the Keycloak address to perform a redirect to the Login page.
/theme/dhus_theme/account/messages/messages_en.properties	logoutUrl	It reports the Keycloak address to perform the logout.
/theme/dhus_theme/email/messages/messages_en.properties	passwordResetSubject	These parameters allow to personalize the email message sent in case of forgot password.
	passwordResetBody	
	passwordResetBodyHtml	

Table 5: Useful Keycloak configurable parameters

Some of the above listed parameters are not fully set in the provided theme and you need to configure them according to your Operational needs. In particular, the following ones need additional actions:

- *logoutUrl* : Insert here the address of your Keycloak instance and the Realm name.
- *accountRedirectUrl* : Insert here the address of your Keycloak instance and the Realm name.
- *hubName* : Insert here the name of your DHuS instance.
- Adapt all parameters related to the email/ folder with the Service provided by you.

In case you want to personalize in a different way your Keycloak instance, please refer to the official Website [AD-1] for more details.

3.2.2 Keycloak Console Configuration

3.2.2.1 First access

Accessing for the first time the Keycloak console via Web:

https://<ADDRESS>/auth/

the creation of an Administrator User is requested. With this credentials, full permissions to manage all parts of Keycloak are guarantee.

According to the Keycloak installation mode, the Administrator User creation is bit different.

3.2.2.1.1 Installation with embedded DB

At the first login, the following panel appears:

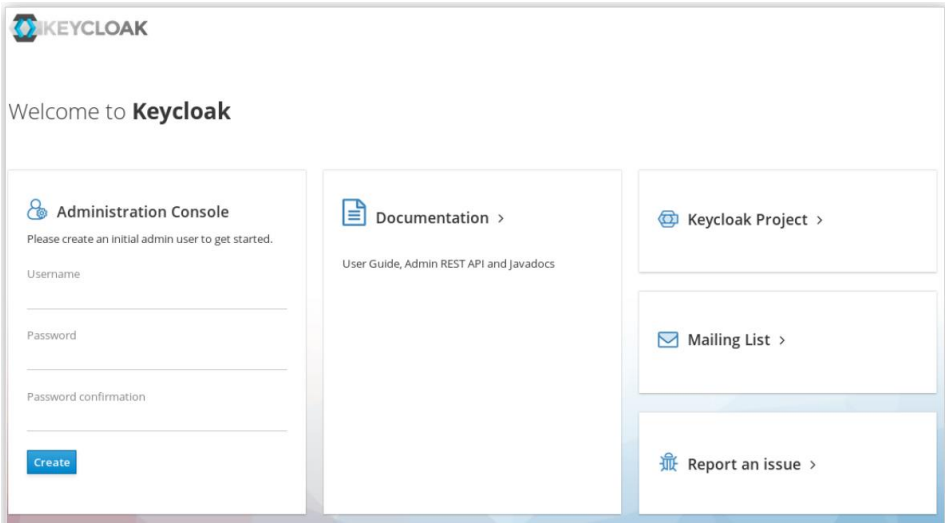

Figure 1: Keycloak First Login

Inserting Username and Password, a User able to access the administration console is created.

3.2.2.1.2 Installation with externalized DB

In order to create an Administrator User, access the bin/ folder and run the following script:

```
./add-user-keycloak.sh -u <username>
```

where <username> is the username of the User you want to create.

As consequence, the setup of User password is required.

At the end of the script run, a User able to access the Keycloak administration console is created.

3.2.2.2 How to interface DHuS instance

In order to let DHuS instances properly interface Keycloak, the following configurations should be set.

Please note that some setups follow the ones did on DHuS side according to [AD-2].

Client creation and configuration

If you need to define more than one client (i.e. you need to register more than one DHuS instances in the same Realm), all Steps involving client creation and configuration have to be repeated for each needed client.

1) On upper left of the console, hover the Realm selection drop-down menu and click on 'Add realm'. Insert the same name configured in the start.sh DHuS configuration file:

   `-Ddhus.saml.idp.name`

2) Access the 'Clients' section and click on 'Create'.
3) Configure:
   - Client ID as the one set in the start.sh `-Ddhus.saml.sp.id`.
   - Client Protocol as `saml`.

   Click on 'Save'.
4) Edit the just created client setting:
   - As Valid Redirect URIs the external address (host + path) defined in the dhus.xml DHuS configuration file. The address should end with '/*'.
   - As Base URL the external address (host + path) defined in the dhus.xml DHuS configuration file.
   - As Master SAML Processing URL the following:

     `<external_address>/saml/saml`
   - In the subsection 'Fine Grain SAML Endpoint Configuration', insert the following address as Logout Service POST Binding URL:

     `<external_address>/saml/saml/SingleLogout`
   - Click on 'Save'.

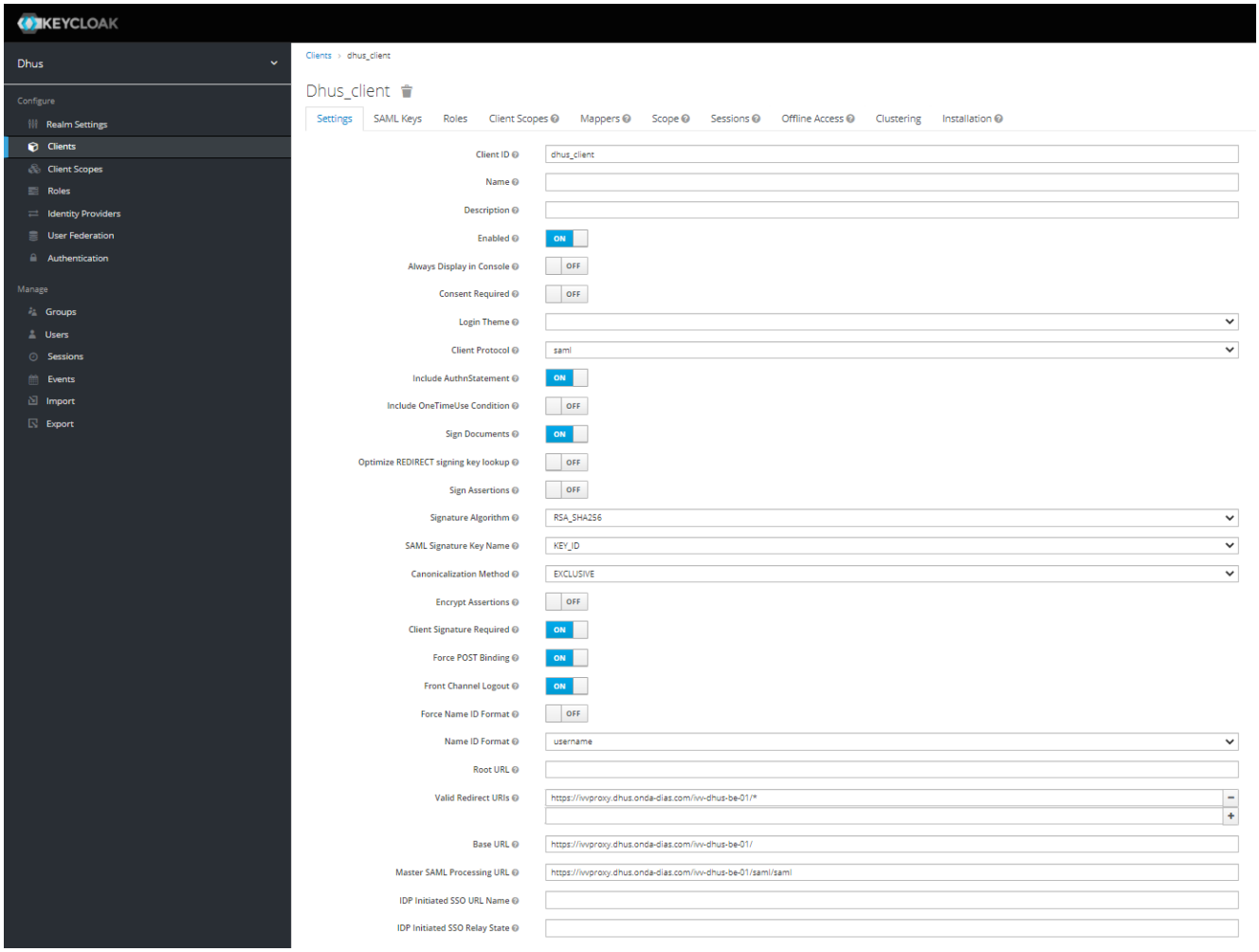

Figure 2: Client Settings

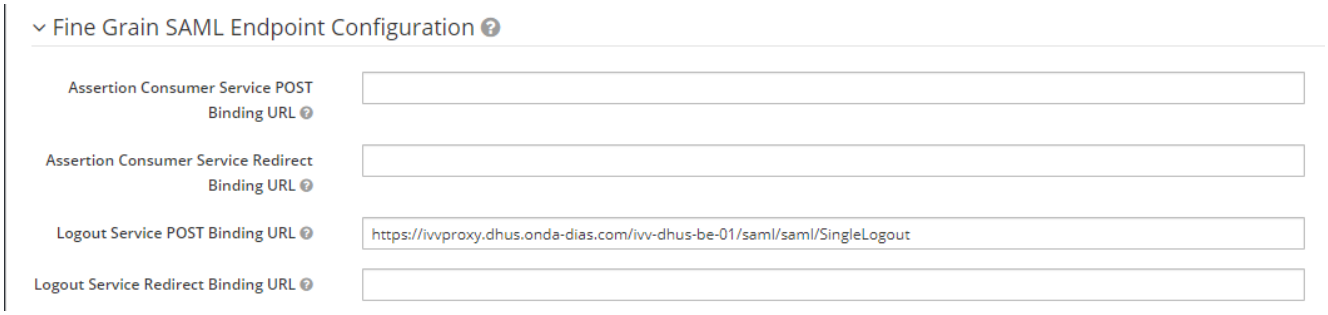

Figure 3: Logout Settings

Allow accessing Keycloak User ID by DHuS

1) Access the 'Client Scopes' section.
2) Click on 'Create' and set:
    - Name equal to id

- Protocol equal to saml

Click on 'Save'.

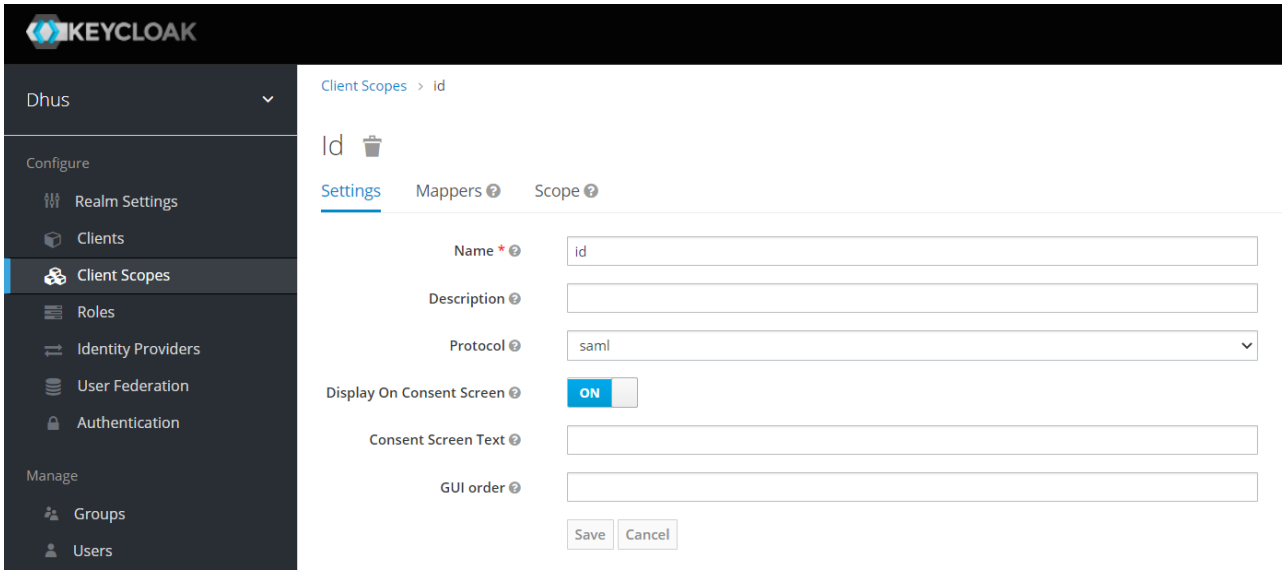

Figure 4: Client Scope creation

3) Access the 'Mappers' tab and click on 'Create'.
4) Set the following:
   - Name = saml
   - Mapper Type = User Property
   - Property = id
   - SAML Attribute Name = samlId
   - SAML Attribute Name Format = Basic

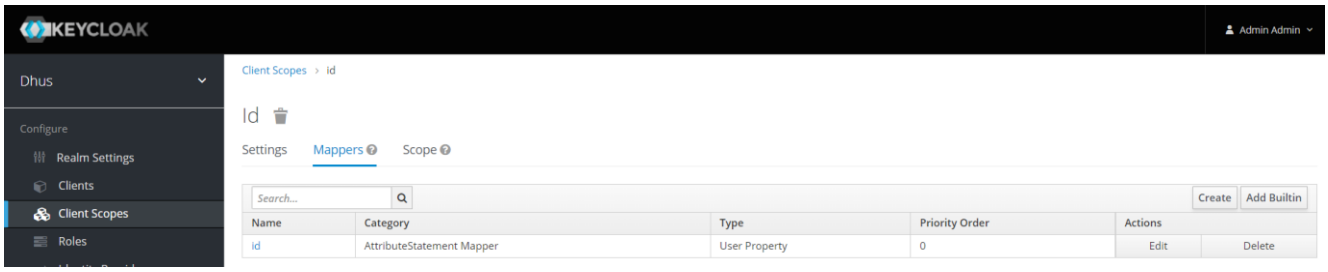

Figure 5: Client Mappers setting

5) Access 'Clients' section.
6) Click on the client ID previously created and access the corresponding 'Client Scopes' tab.
7) Add 'id' to Assigned Default Client Scopes.

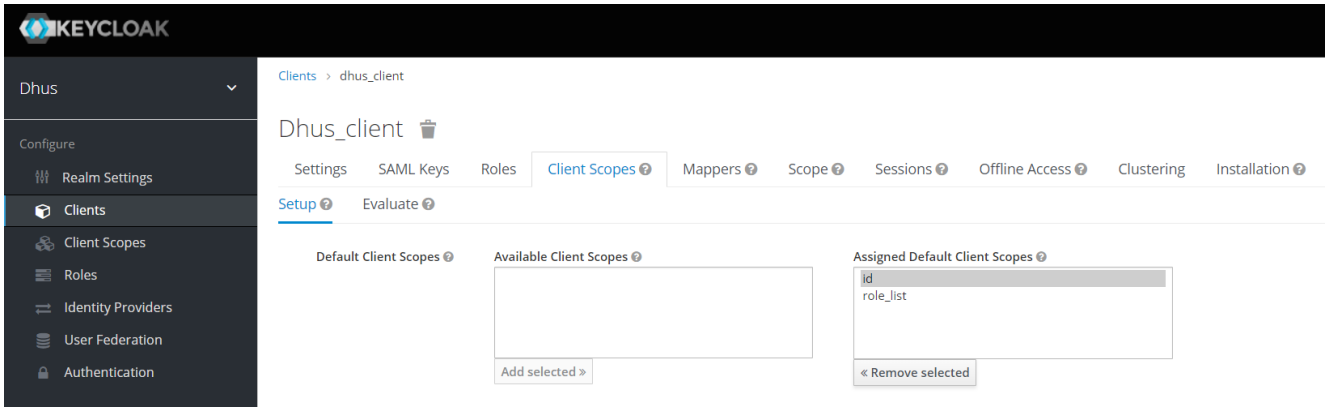

Figure 6: Default Client Scope setting

Importing SAML Key

1) Access the 'Clients' section and select the Client ID previously created.
2) Access the 'SAML Keys' tab and click on 'Import'.
3) Set the following parameters according to what inserted in the command used to generate the key on DHuS side [AD-2]:
   - Key Alias
   - Store Password
4) Click on 'Select file' and insert the 'dhusKeystore.jks' generated on DHuS side.

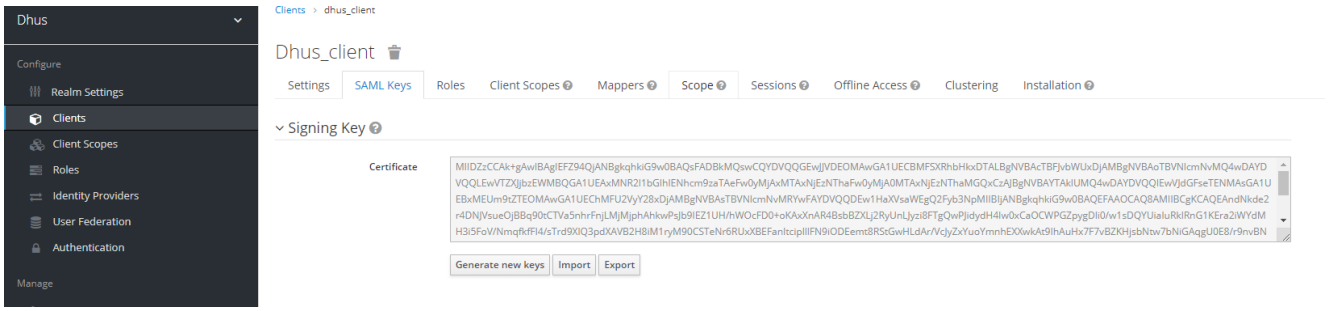

Figure 7: How to import SAML Key

Allow accessing Keycloak User Roles by DHuS

1) Access the 'Clients' section and select the Client ID previously created.
2) Access the 'Roles' tab and click on 'Add Role'.
3) Referring to the table below, add all User Roles supported by the DHuS by inserting their name and description. At the end of the insertion, click on 'Save'.

   Please note that the 'Description' field is optional.

Roles	Description
AUTHED	Default Role. It is required for DHuS internal mechanism.
SEARCH	Default Role. Allow User to search for products in DHuS using all available interfaces and to manage his saved searches.
DOWNLOAD	Default Role. Allow User to manage his product cart.
UPLOAD	Allow User to upload products, via manual upload, OData or scanner (if he is also Data Manager).
USER_MANAGER	Allow User to manage other Users, including create/update/delete/lock/unlock accounts.
DATA_MANAGER	Allow User to delete Products, DeletedProducts and to manage Collections, Scanners.
SYSTEM_MANAGER	Allow User to see Network information, to manage DataStores, Evictions, Orders, Scanners, Synchronizers, Transformations and to repair Products via OData.
ARCHIVE_MANAGER	Allow User to see LocalPath of Products via OData v1, used in Synchronization.

Table 6: DHuS User Roles

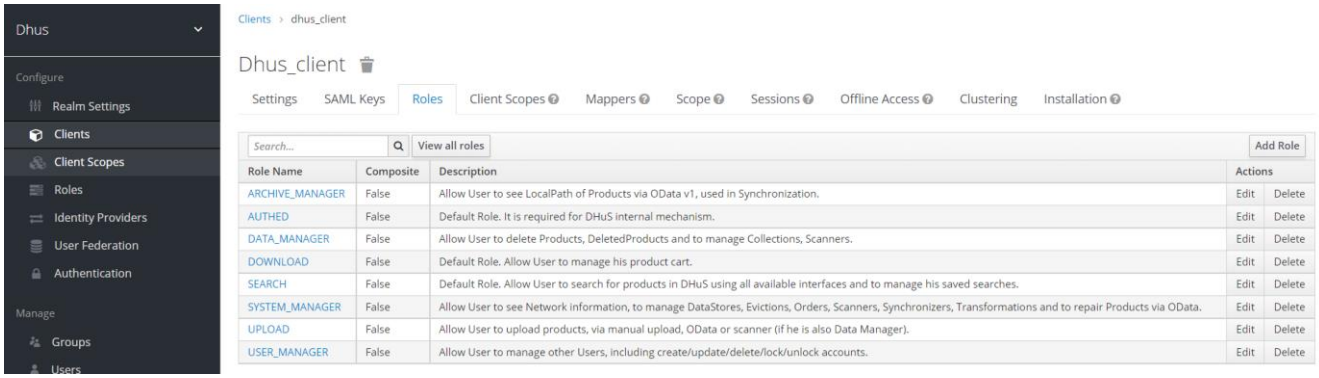

Figure 8: User Roles configuration

Define default roles for each new registered User

1) Access the 'Roles' section.
2) Access the 'Default Roles' tab.
3) In the drop-down menu 'Client Roles', select the Client ID previously created.
4) In the table 'Available Roles', select the default one to be assigned and click on 'Add selected'.

   Please note that the default User Role on DHuS SW are SEARCH and DOWNLOAD.

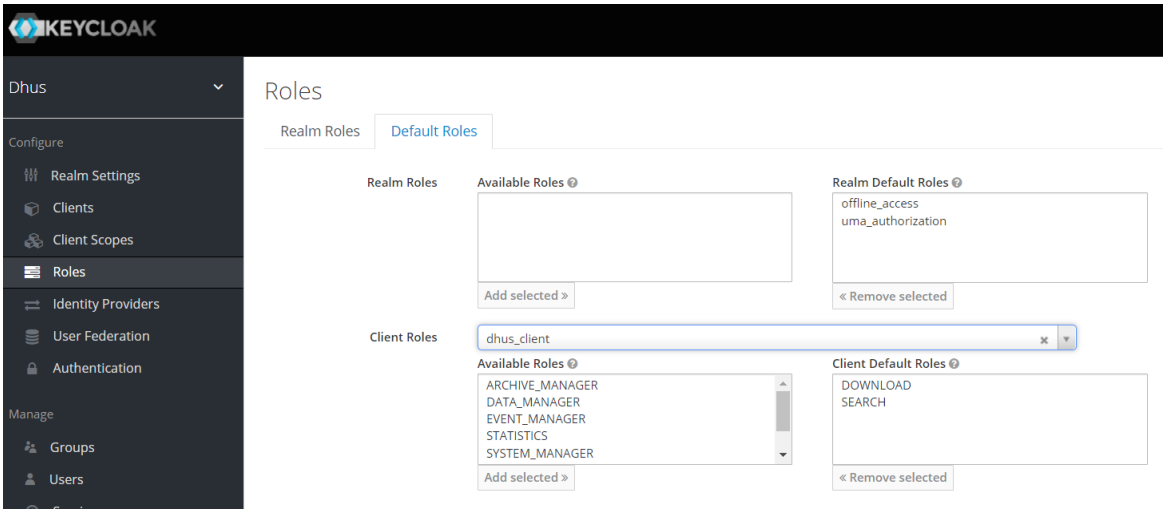

Figure 9: Default User Role definition

Avoid receiving User Roles from other clients

1) Access the 'Clients' section and select the Client ID previously created.
2) Access the 'Scope' tab and put on OFF the 'Full Scope Allowed' parameter.

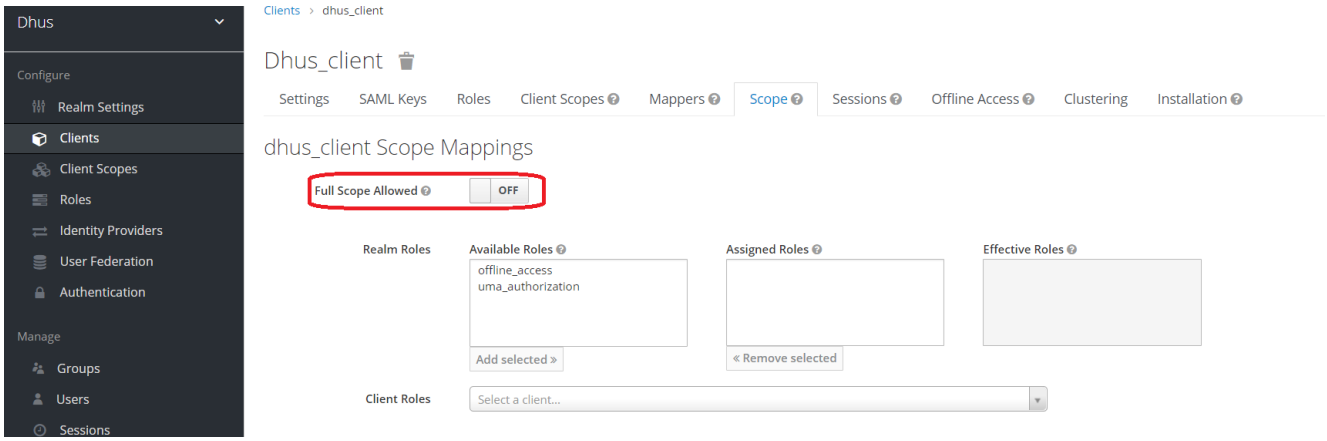

Figure 10: How to avoid receiving User Roles from other clients

3.2.2.3 How to setup Password Policy

According to the password policies applied to DHuS SW, a User can choose his/her password by satisfying the following minimal requirements:

- Password field accepts only alphanumeric characters plus "!", "@", "#", "$", "%", "^", "&", "*", ")", "(", "+", "=", ".", "_", "-".
- Password fields minimum length is 8 characters.

In order to ensure these restrictions also during password setup via Keycloak, the following steps should be performed:

1) Access the 'Authentication' section.
2) Access the 'Password Policy' tab.
3) From the drop-down menu 'Add policy…', select 'Regular Expression'. In the corresponding 'Policy Value' field insert the following expression:

   ^[a-zA-Z0-9!@#\$%\^\&*\)\(+=._-]+$

   Click on 'Save'.
4) From the drop-down menu 'Add policy…', select 'Minimum Length'. In the corresponding 'Policy Value' field insert 8. Then click on 'Save'.

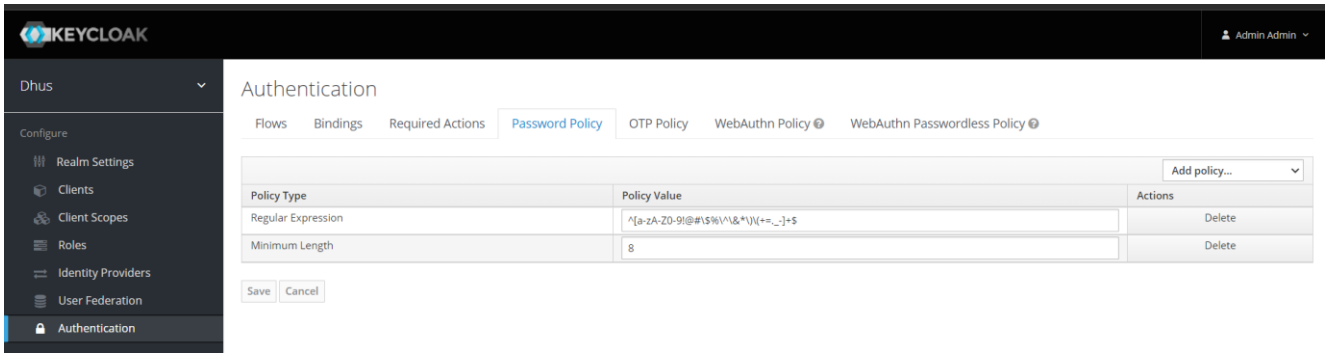

Figure 11: Password Policy setup

3.2.2.4 How to setup SMTP Service

When a User asks for a 'Forgot Password' service, an email is sent to him/her in order to reset his/her own password.

In order to guarantee this in case of 'Forgot Password' requested via Keycloak, a SMTP Service should be configured in the following way:

1) Access the 'Realm Settings' section.
2) Access the 'Email' tab.
3) Fill at least the following fields:
   - Host: It denotes the SMTP server hostname used for sending emails.
   - From: It denotes the address used for the From SMTP-Header for the emails sent.
4) According to the type of the selected SMTP service, activate or not the following buttons:
   - Enable SSL
   - Enable StartTLS
   - Enable Authentication
5) In case of 'Enable Authentication' activate, fill the 'Username' and 'Password' fields that appear with the credentials to access the email address set in the 'From' field.
6) Click on 'Save'.

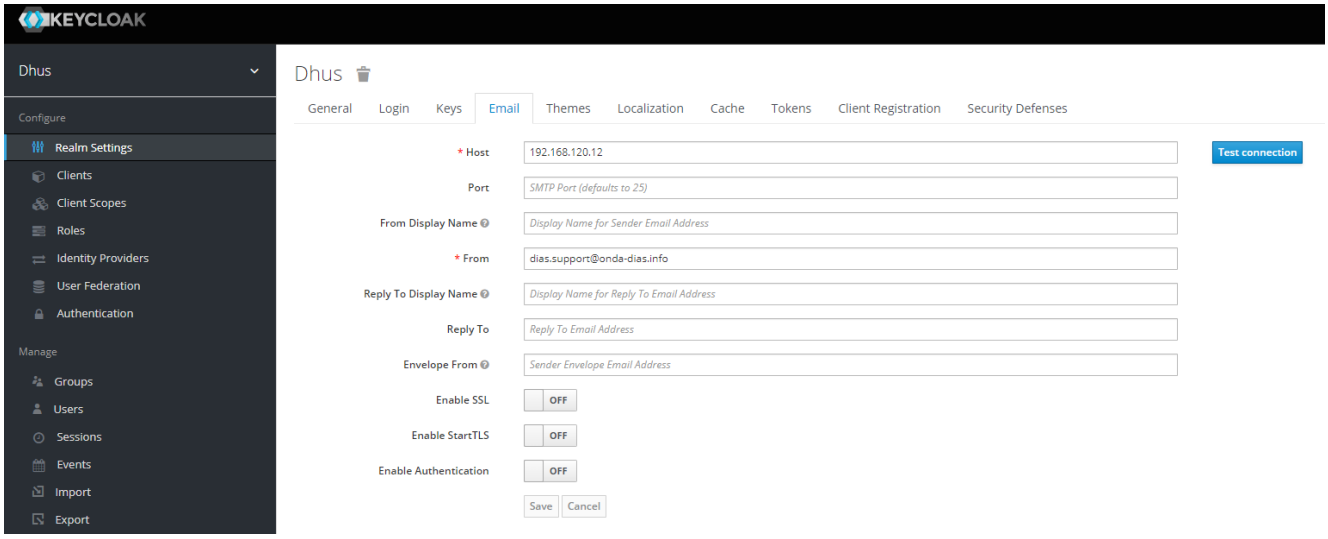

Figure 12: SMTP Service setup

Next step to enable the emails sending is to associate the same mail configured in the 'From' field to the Keycloak Administrator account:

1) Access the 'Manage account' section related to Administrator User accessible via the drop-down menu at the top right of the console.
2) Access the 'Personal info' section.
3) Configure the 'Email' as the email address that will send notifications to the Users.

   Please note that also the fields 'First name' and 'Last name' are mandatory to save the modifications made.
4) Click on 'Save'.

Please note that these additional steps are mandatory to ensure the correct Users emails sending.

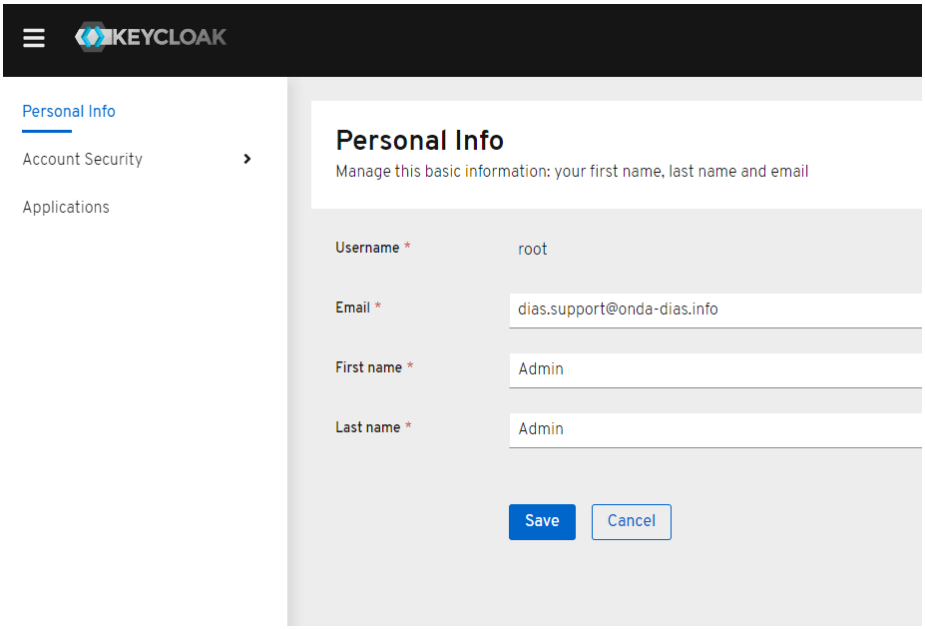

Figure 13: Email Administrator User configuration

It is possible to configure the content of the email sent to the Users by customizing the theme associate to this functionality.

The provided theme has been adapted to send the following mail for allowing the User to reset his/her own password:

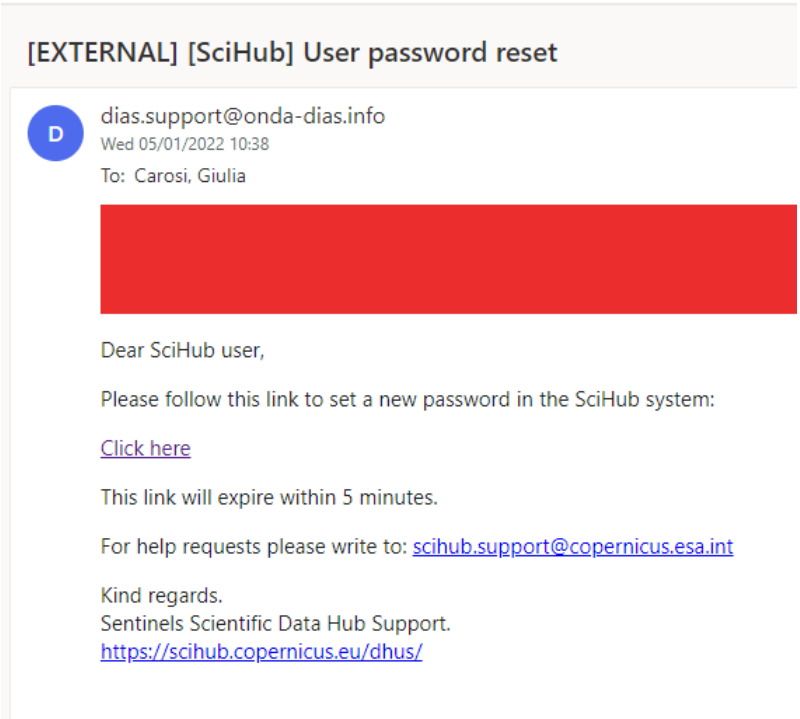

Figure 14: Email Forgot Password

In order to instruct Keycloak to refer to this configuration for the email setup, the following steps should be performed:

1) Access the 'Realm Settings' section.
2) Access the 'Themes' tab.
3) From the drop-down menu corresponding to 'Email Theme', select 'dhus_theme'.

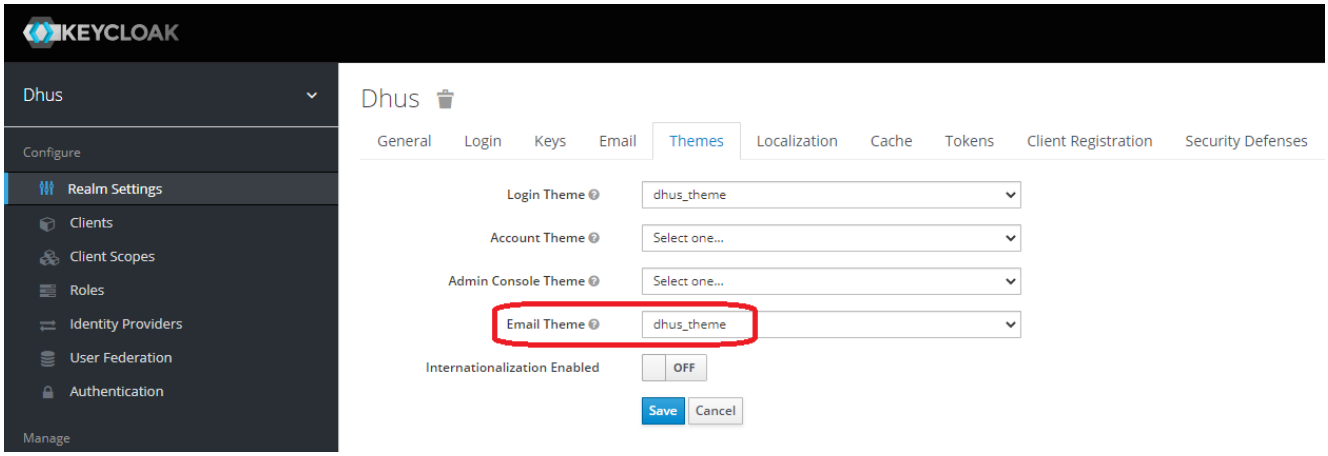

Figure 15: Email theme selection

3.2.2.5 How to setup Login panel

The Keycloak customization made with the provided theme adapts the Keycloak Login panel to the one currently present in the DHuS GUI.

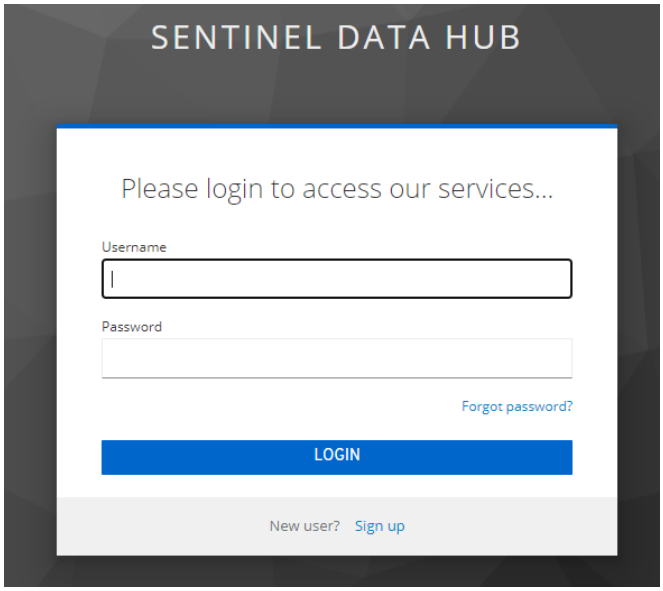

Figure 16: Login panel

In order to activate this configuration for the Login panel, the provided theme should be select for the Login functionality. It can be done with the following steps:

1) Access the 'Realm Settings' section.
2) Access the 'Themes' tab.
3) From the drop-down menu corresponding to 'Login Theme', select 'dhus_theme'.
4) Click on 'Save'.

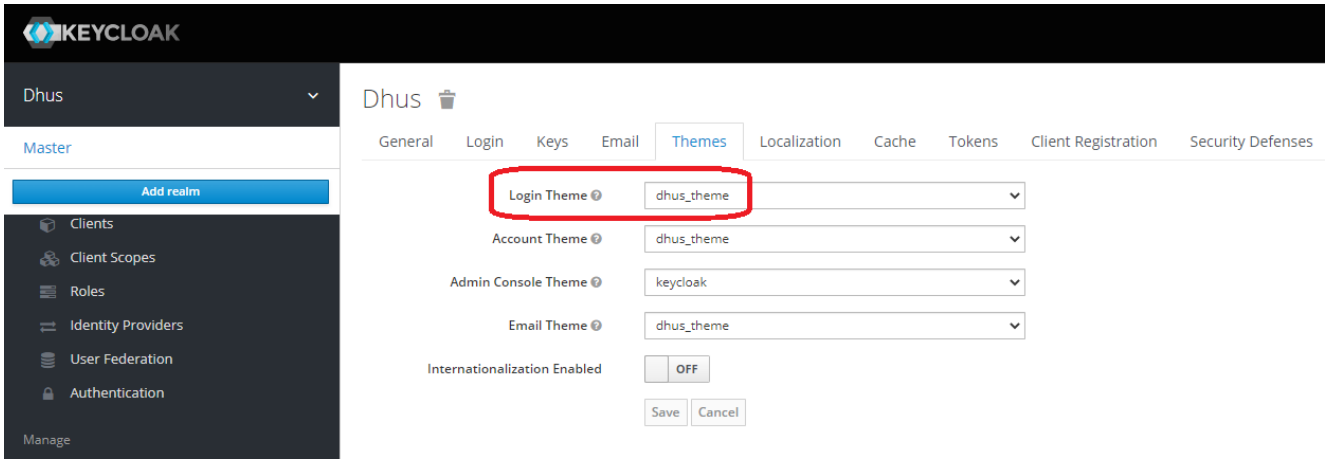

Figure 17: Login theme selection

3.2.2.6 How to setup Edit Profile panel

After registered on a DHuS instance, each User can edit his/her own profile by modifying all fields except the 'Username' one.

With the provided theme, the 'Edit Profile' panel will appear as shown below:

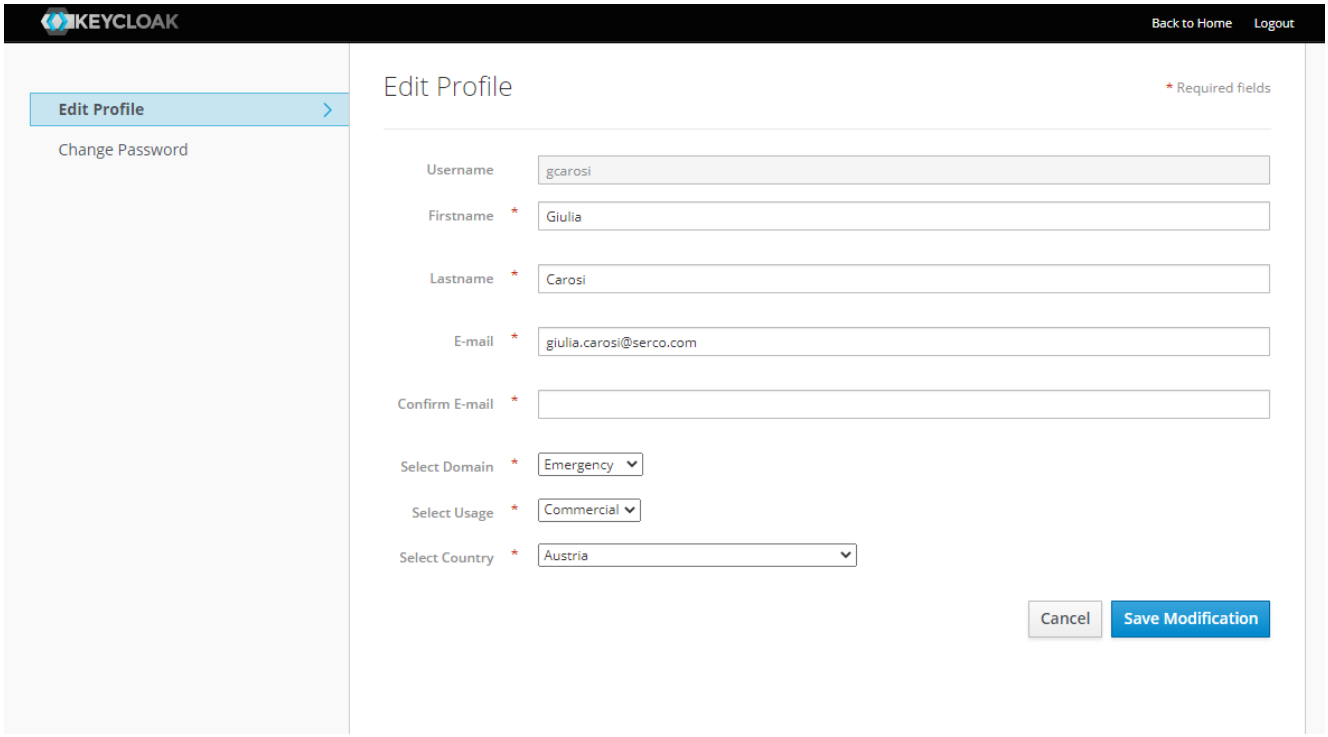

Figure 18: Edit Profile panel

In order to activate this configuration for the 'Edit Profile' panel, the provided theme should be select for the Account functionality. It can be done with the following steps:

1) Access the 'Realm Settings' section.
2) Access the 'Themes' tab.
3) From the drop-down menu corresponding to 'Account Theme', select 'dhus_theme'.
4) Click on 'Save'

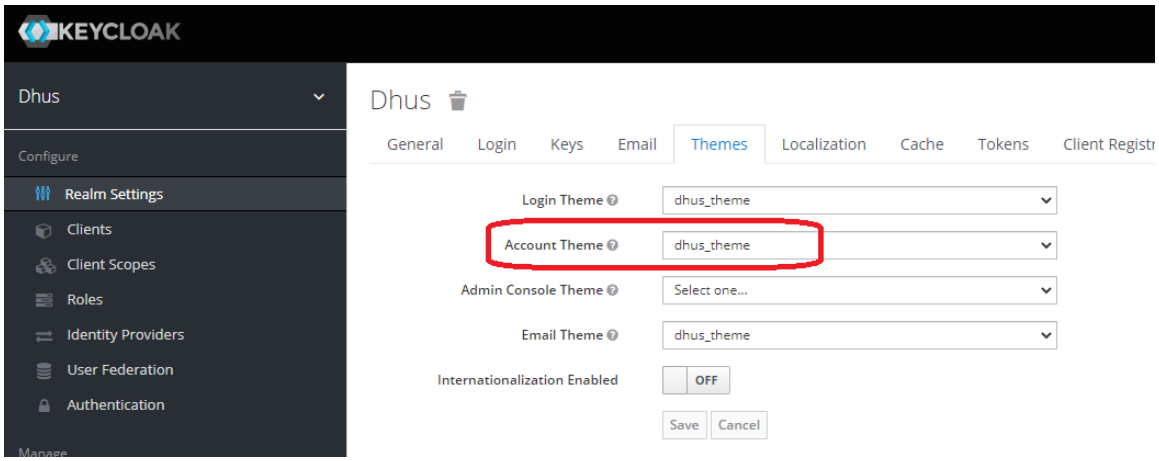

Figure 19: Edit Profile theme selection

3.2.2.7 How to setup Forgot Password panel

By default, Keycloak does not allow the Forgot Password for its own registered Users.

In order to activate this functionality, the following steps should be executed:

1) Access the 'Realm Settings' section.
2) Access the 'Login' tab and enable the 'Forgot password' button.
3) Click on 'Save'.

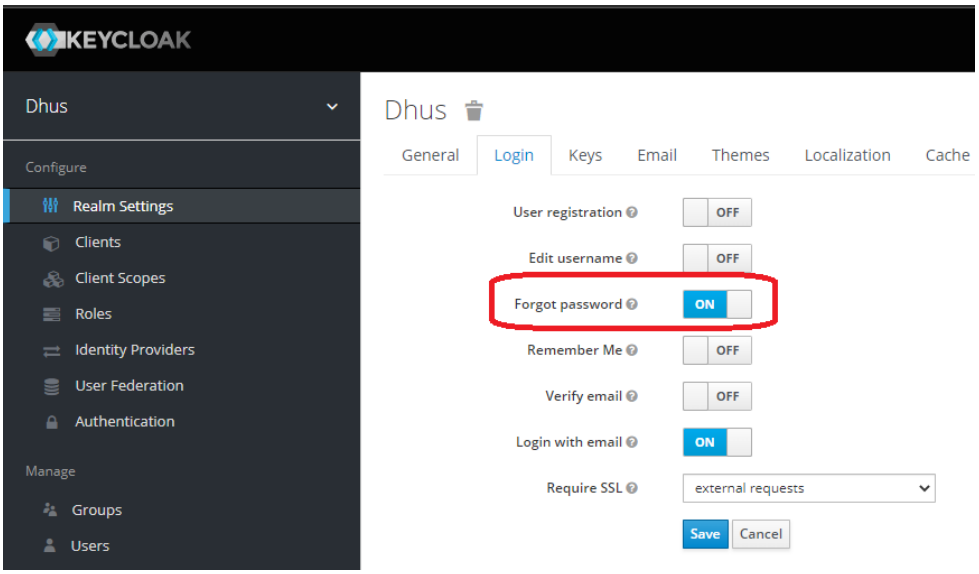

Figure 20: Forgot Password activation

With the provided theme, the 'Forgot Password' panel will appear as shown below:

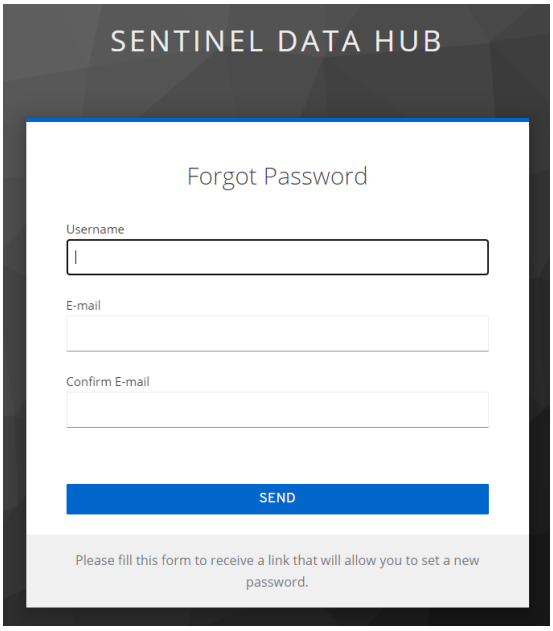

Figure 21: Forgot Password panel

In order to activate this configuration for the Forgot Password panel, the provided theme should be select for the Login functionality. It can be done with the following steps:

5) Access the 'Realm Settings' section.
6) Access the 'Themes' tab.
7) From the drop-down menu corresponding to 'Login Theme', select 'dhus_theme'. Please refer to Figure 17: Login theme selection for more details.

3.2.2.8 How to setup Self Registration panel

As User Management provider, Keycloak should allow the Registration of Users that want to access and explore the DHuS instance.

The access to the Keycloak Registration form can be enabled by adding a link in the Login panel. The following steps describe how to do this:

1) Access the 'Realm Settings' section.
2) Access the 'Login' tab and enable the 'User registration' button.
3) Click on 'Save'.

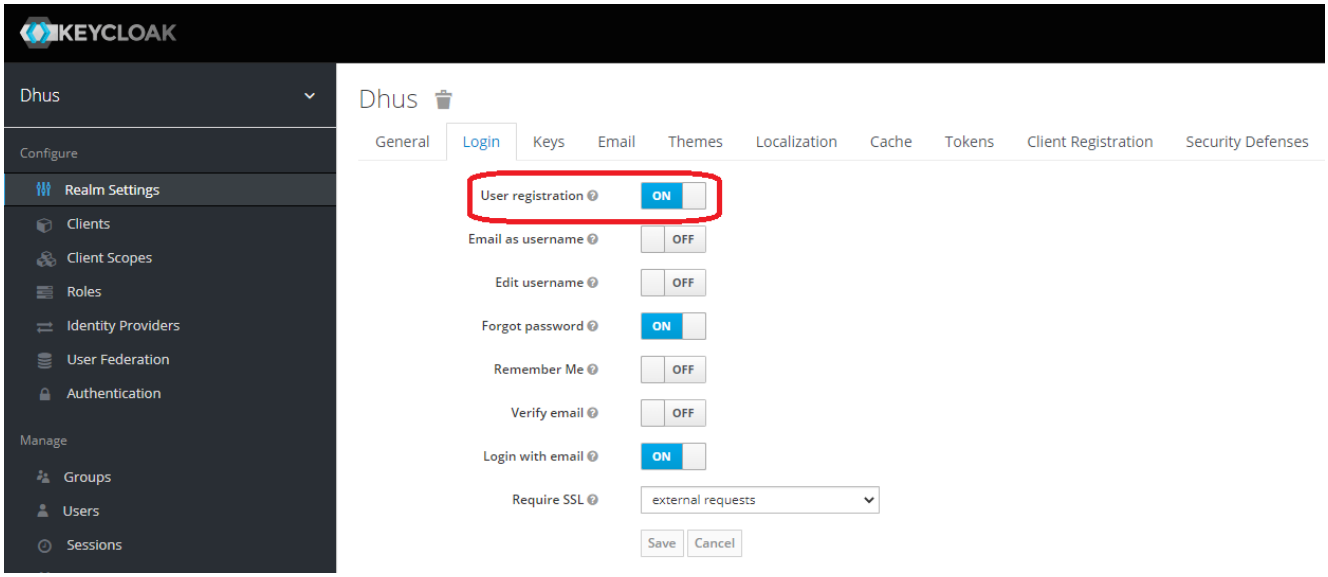

Figure 22: Link to Registration form setup

By default, Keycloak allows the User Login with email and no duplicate emails for registered Users. On DHuS side, the expected behaviour is the opposite one. To change these Keycloak setups, the following steps should be executed:

1) Access the 'Realm Settings' section.
2) Access the 'Login' tab and disable the 'Login with email' button.
3) Enable the 'Duplicate emails' button that appears after the previous action.
4) Click on 'Save'.

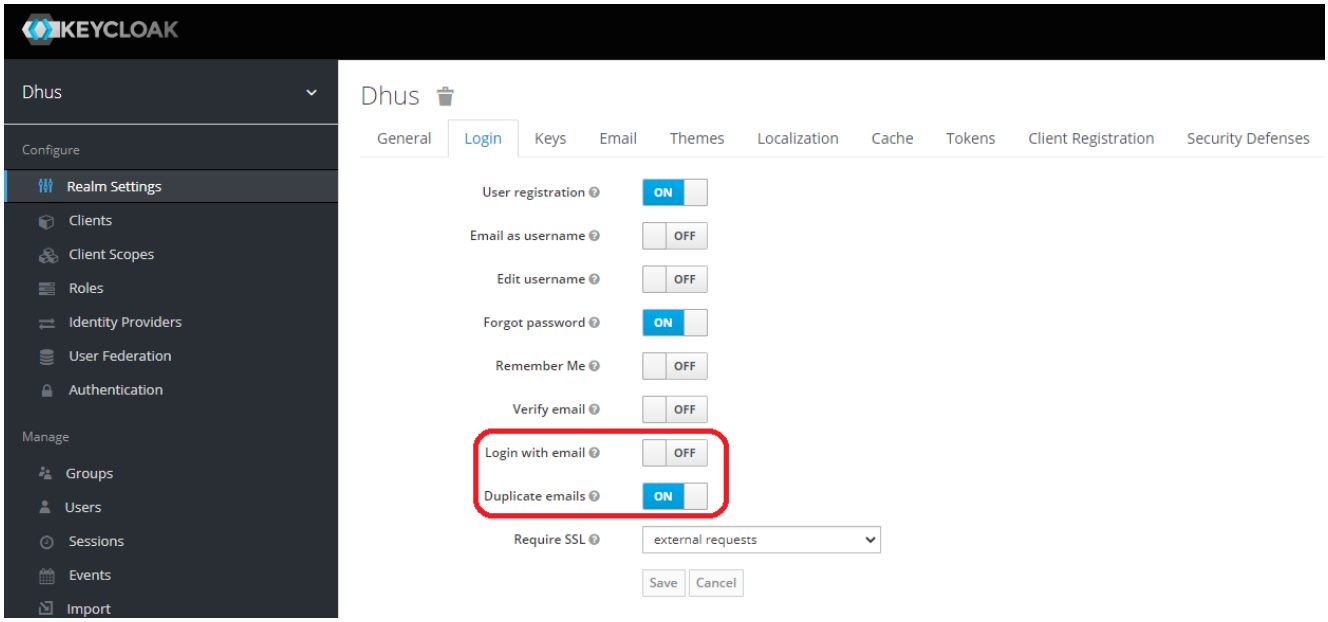

Figure 23: Allow duplicated emails for registered Users

The Figure below shows the Self Registration panel as implemented in the provided theme.

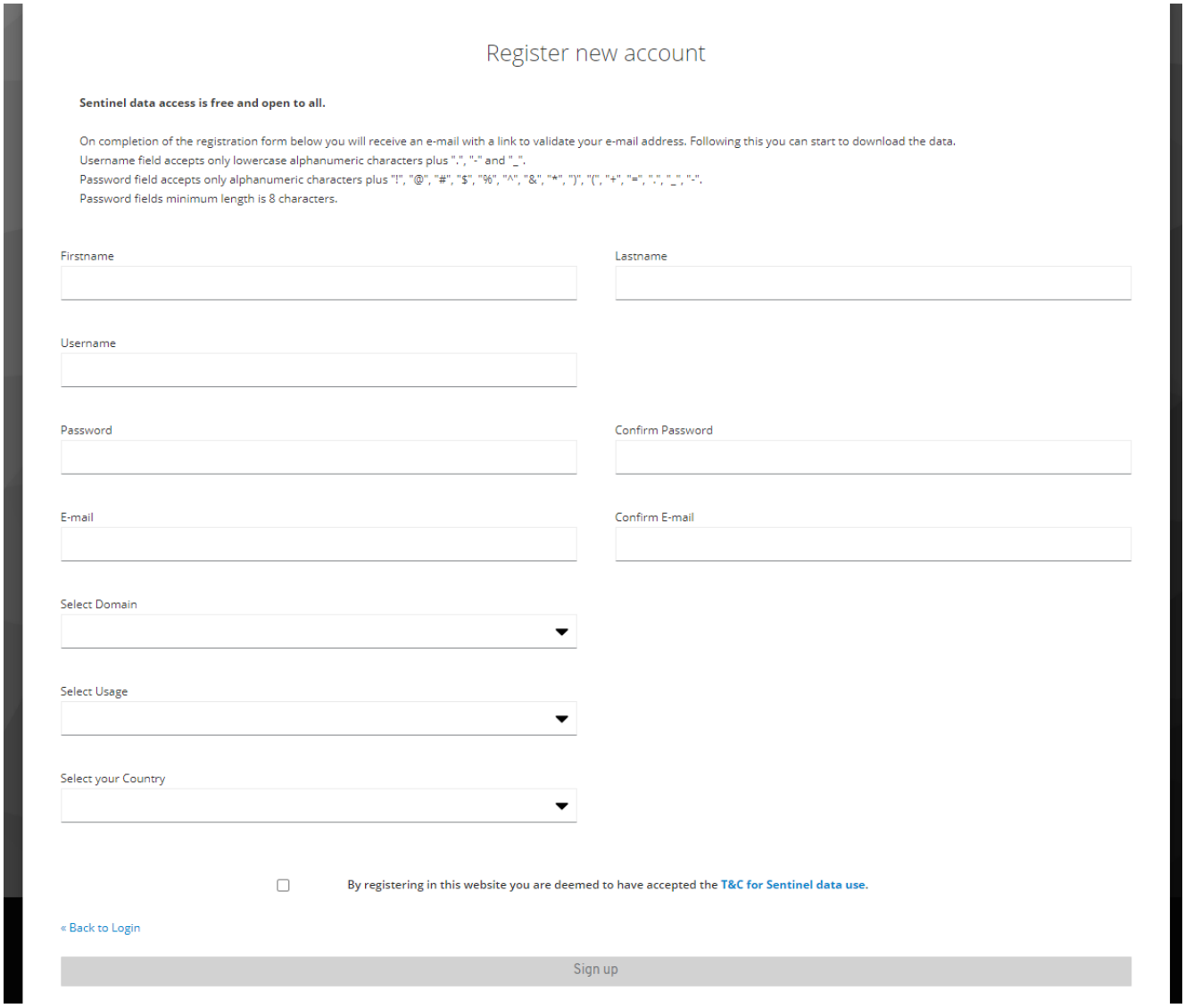

Figure 24: Self Registration panel

Please note that the link 'T&C for Sentinel data use' redirects to the page:

https://scihub.copernicus.eu/twiki/do/view/SciHubWebPortal/TermsConditions

In order to activate this configuration for the Self Registration panel, the provided theme should be select for the Login functionality. It can be done with the following steps:

1) Access the 'Realm Settings' section.
2) Access the 'Themes' tab.
3) From the drop-down menu corresponding to 'Login Theme', select 'dhus_theme'. Please refer to Figure 17 for more details.

By default, Keycloak automatically login the User after the registration. For a correct behaviour of the DHuS SW this should be avoided.

In order to do this, the following actions should be executed:

1) Access the 'Authentication' section.
2) From the drop-down menu corresponding to the available 'Flows', select 'Registration'.
3) Click on 'Copy'.
4) As New Name, insert 'Self Registration'. Then click on 'Ok'.
5) Access the new created flow and click on 'Add execution'.
6) In the drop-down menu 'Provider', select 'Username Password Form'. Click on 'Save'.
7) Mark as REQUIRED.

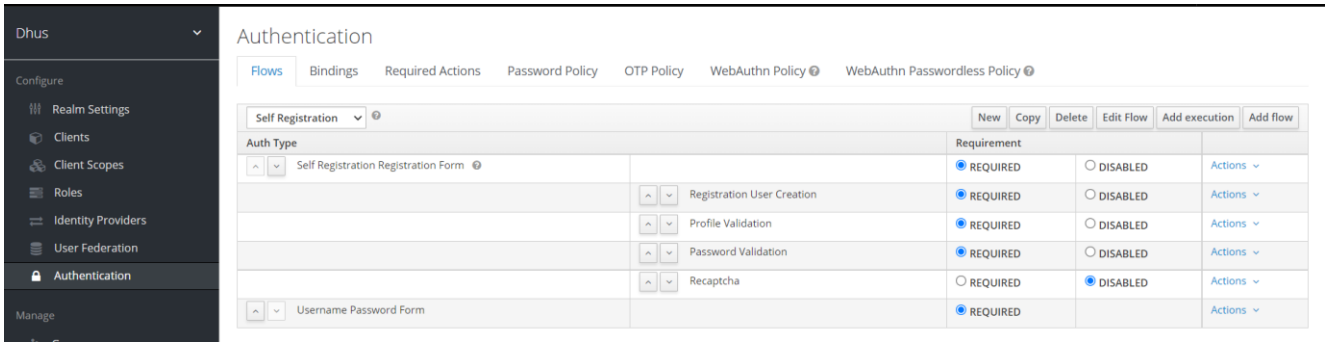

Figure 25: Flow to avoid Login after Registration

3.3 User migration from a DHuS instance

As highlighted in the previous Chapters, DHuS 3.0.X branch has been adapted to be compliant with GDPR requirements.

The main consequence is that the User management is fully redirected to Keycloak and no more User information are stored at DHuS level.

To cope with this, a script to delete all User information from DHuS instance is provided within DHuS 3.0.X distributions. Please refer to [AD-2] for more details on the script usage.

For those who want to install DHuS 3.0.X version and activate the GDPR, this Chapter describes how to migrate all Users already registered in the DHuS instance to Keycloak.

Please note that the User migration shall be executed before the run of the script to delete all Users.

3.3.1 Enable MD5 as password encrypt algorithm

DHuS SW stores Users passwords in its own Database by encriptying them with a MD5 algorithm.

In order to ensure that Keycloak correctly manage these encrypted passwords when Users are migrated, a .jar file has been developed:

serco-md5-crypt.jar

Please refer to relevant DHuS release notification to download it.

The script customizes Keycloak to accept MD5 as encryption algorithm for stored password. In order to allow this, the .jar file has to be added in the Keycloak configuration folders:

1) Stop Keycloak instance.
2) Unzip the provided file and put the .jar file in the standalone/deployments/ folder.
3) Start Keycloak and check that the file is listed in the server.log application log file.

INFO [org.jboss.as.server.deployment] (MSC service thread 1-1) WFLYSRV0027: Starting deployment of "serco-md5-crypt.jar" (runtime-name: "serco-md5-crypt.jar")

INFO [org.keycloak.subsystem.server.extension.KeycloakProviderDeploymentProcessor] (MSC service thread 1-8) Deploying Keycloak provider: serco-md5-crypt.jar

INFO [org.jboss.as.server] (ServerService Thread Pool -- 33) WFLYSRV0010: Deployed "serco-md5-crypt.jar" (runtime-name : "serco-md5-crypt.jar")

Next step is to add this new password policy in the dedicated section in the Keycloak administration console related to the Realm created in Section 3.2.2.2.

Please note that this change shall be performed without any registered Users on the Realm, otherwise they will no more be able to access it.

1) Access the 'Authentication' section.
2) Access the 'Password Policy' tab.
3) From the drop-down menu 'Add policy…', select 'Hashing Algorithm'. In the corresponding 'Policy Value' field insert the following expression:

   md5crypt

   Click on 'Save'.

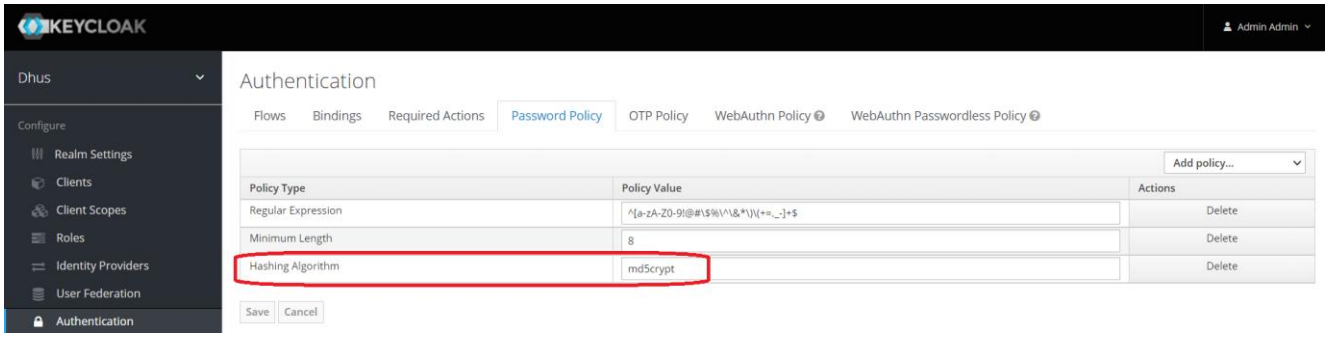

Figure 26: MD5 encryption algorithm setup

3.3.2 How to import User

In order to import Users registered on a DHuS instance in the configured Keycloak instance, a Python script

dhus_users_migrator.py

has been developed. Please refer to relevant DHuS release notification to download it.

Please refer to the following Sections for more details on its usage.

3.3.2.1 Pre-requirements

The following preconditions shall be satisfied:

- The script shall run in a Linux environment;
- Python v3.7 shall be installed;
- At least, the following Python modules shall be installed:
  - python-keycloak v0.26.1
  - urllib3 v1.26.7
- Keycloak Java memory shall be set as follows:
  - Xms512m
  - Xmx1024m

  Please note that it is possible to configure these parameters in the bin/standalone.conf Keycloak configuration file.

For your best convenience, please find below an additional list of Python package installed in the VM where the script has been developed and that guarantee a correct script run:

- certify v2021.10.8
- charset-normalizer v2.0.9
- ecdsa v0.17.0
- idna v3.3
- pyasn1 v0.4.8
- python-jose v3.3.0
- requests v2.26.0
- rsa v4.8
- six v1.16.0

If you encounter some issue at the script start, please check also these libraries' versions.

3.3.2.2 Realm backup

To prevent any loss of data, the export of the Realm previously configured is recommended:

1) Access the 'Export' section.
2) Enable 'Export groups and roles' and 'Export clients'.
3) Click on 'Export'.

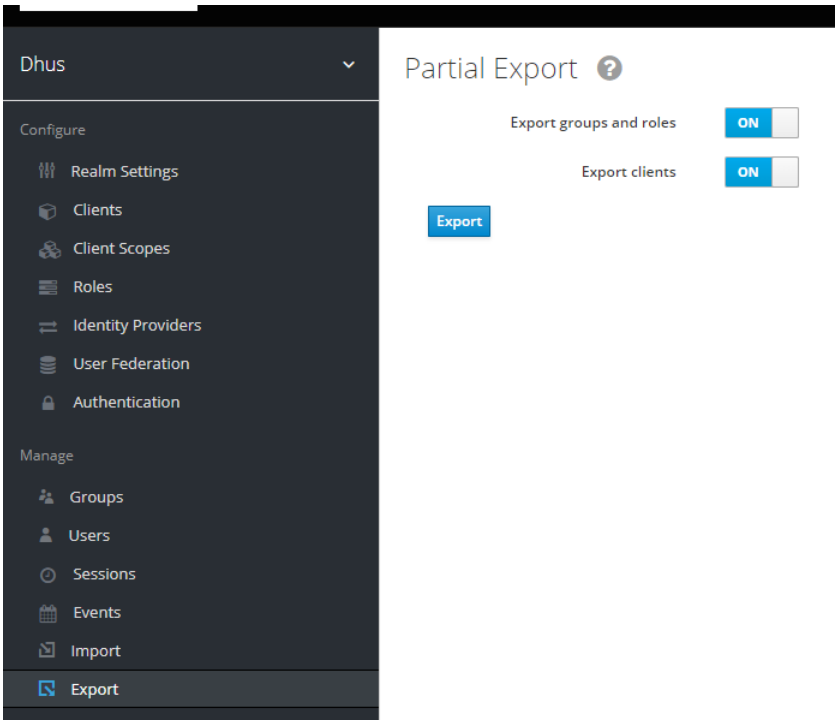

Figure 27: How to export a Realm

As result, a JSON file containing all info related to the exported Realm is downloaded.

3.3.2.3 User migration script usage

The User migration script takes as inputs:

- csv_path: the path of a csv file containing the export of the 'users' table stored in the DHuS DB.

  In the Section 3.3.2.3.1, details on how the csv file should be provided are given.
- keycloak_host_port: the Keycloak instance address in the format <protocol>://<address>:<port>.
- keycloak_admin_user: the username of the Administrator User created at the first Keycloak login.
- keycloak_admin_password: the password of the Administrator User created at the first Keycloak login.
- keycloak_import_realm: the Keycloak Realm name on which Users have to be migrated.

In order to set these parameters, edit the User migration script and then save the changes made.

In addition to the above listed parameters, the following one shall be set to True:

- verify_ssl = True

The script can be run with the following command:

```
python3 dhus_users_migrator.py
```

In case of a large number of Users to migrate, the 'nohup' command is suggested.

The output of the script are two csv files:

- inserted_users.csv: each line reports the Keycloak User's uuid associated to the successfully inserted User;
- errors_csv: in case an error occurred during the User migration, a line is reported in this file showing the associated User's uuid.

If any error occurs during the Users migration, it is possible to rollback to the Realm configuration before the run of the script performing the following actions:

1) Delete the Realm by clicking on the recycle bin icon near the Realm's name

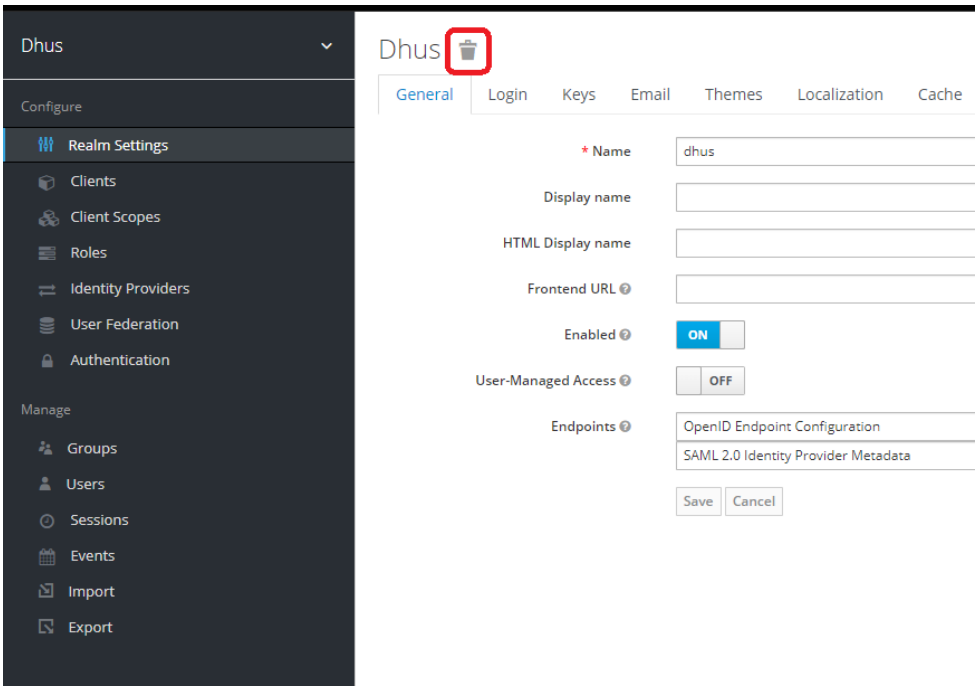

Figure 28: How to delete a Realm

2) On upper left of the console, hover the Realm selection drop-down menu and click on 'Add realm'. Click on 'Select file' and import the JSON file related to the Realm previously downloaded.
3) Click on 'Create'.

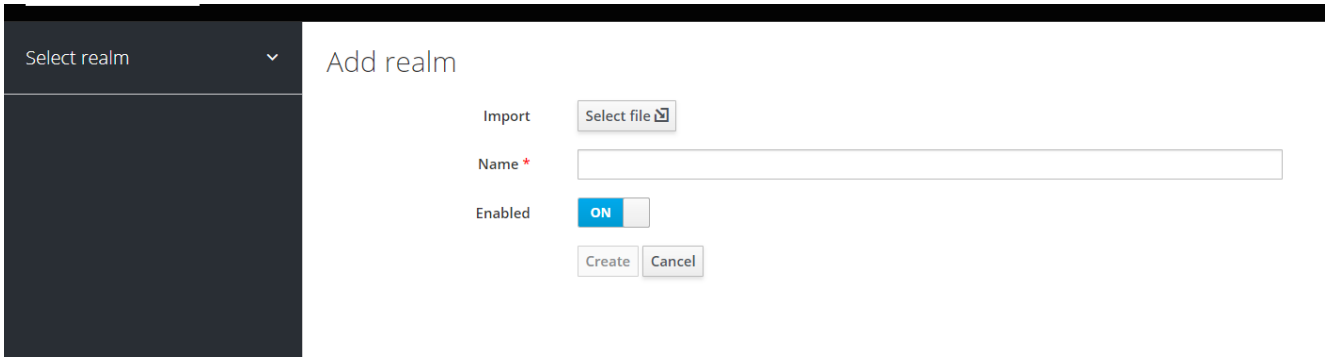

Figure 29: How to import a Realm

3.3.2.3.1 Notes

Please be aware of the following notes during the User migration:

- If you have more than one client defined in the same Realm, please be sure to create the clients before the User Migration. This ensures that the Default Roles selected in Section 3.2.2.2 will be applied to all clients.
- Please note that migrated Users will have assigned only Default Roles previously selected. If a User has additional role on DHuS side before the migration, it has to be manually updated by adding the involved new Role.
- Please note that Keycloak is case insensitive meaning that Users' usernames that contains upper case letters will be migrated with all lower case letters.

  This also causes the not migration of Users that have on DHuS side the same username that differs only for some upper case letter. For example, if two Users have 'test' and 'Test' as username on DHuS side, only the User 'test' will be migrated and 'Test' will result as error as output of the migration script.

3.3.2.4 How to export Users info from DHuS DB

The User script migration requires as input a csv file containing an export of the 'user' table of DHuS DB.

Please note that the provided csv file shall contain Users info in the following order:

country, usage, email, firstname, lastname, password, domain, login

The following sub-sections give details on how to generate this csv file according to the DHuS installation mode.

3.3.2.4.1 DHuS embedded DB

The Table below summarizes the steps to interrogate a DHuS embedded DB and extract the 'user' table in a csv file.

Step ID	Actions
1	On the VM where DHuS is installed, download the provided folder query_hsqldb.zip. Please refer to relevant DHuS release notification to download it.

2	Unzip the downloaded folder: > unzip query_hsqldb.zip
3	Stop DHuS instance.
4	Execute the following command to generate the csv file: > echo "select country, usage, email, firstname, lastname, password, domain, login from user;" | /path/to/query_hsqldb.sh -m 20g -d /path/to/var/database/ --dhus-core /path/to/dhus-core.jar --no-crypt > /path/to/file.csv where: • */path/to/query_hsqldb.sh* is the entire path where the script query_hsqldb.sh is located; • */path/to/var/database/* is the path to the embedded DB w.r.t the DHuS installation folder; • */path/to/dhus-core.jar* is the path to DHuS core, i.e. the lib/ folder inside the DHuS installation folder; • */path/to/file.csv* is the path where to download the csv file.
5	From the generated csv file, remove the first three lines.

3.3.2.4.2 DHuS externalized DB

To interrogate a DHuS externalized Postgres DB and extract the 'user' table in a csv file, the following query should be executed from the DB:

\copy (select country, usage, email, firstname, lastname, password, domain, login from users) To '/path/to/file.csv' With CSV DELIMITER '|';

where */path/to/file.csv* is the path where to download the csv file.

4. Transformation Framework

In the frame of the Transformation Framework DHS component, the Identify and Access Management is delegated to Keycloak software.

Next Sections describe how to configure Keycloak to be compliant with actual Transformation Framework for all concerns User Management actions.

4.1 Keycloak Console Configuration

4.1.1 Realm creation

In order to create needed clients, a Realm in Keycloak is needed. The Realm can be created as follows:

- On upper left of the console, hover the Realm selection drop-down menu and click on 'Add realm'.

Note: In case a Realm is already existing (e.g. DHuS realm), the same Realm could be used to host the Transformation Framework client, sharing the same Users.

4.1.2 Clients' creation and configuration

Two different clients are needed to integrate the Transformation Framework with Keycloak:

- A "Public" client to request and obtain an Access Token
- A "Bearer-Only" client to validate the obtained Access Token

In order to create the requested clients, follow the following steps.

Public Client creation

1) Access the 'Clients' section and click on 'Create'.
2) Configure:
    - Client ID = Name of the public client (e.g. esa_tf_client_public_login)
    - Client Protocol = openid-connect

   Click on 'Save'.
3) Edit the just created client setting:
    - Access Type = public

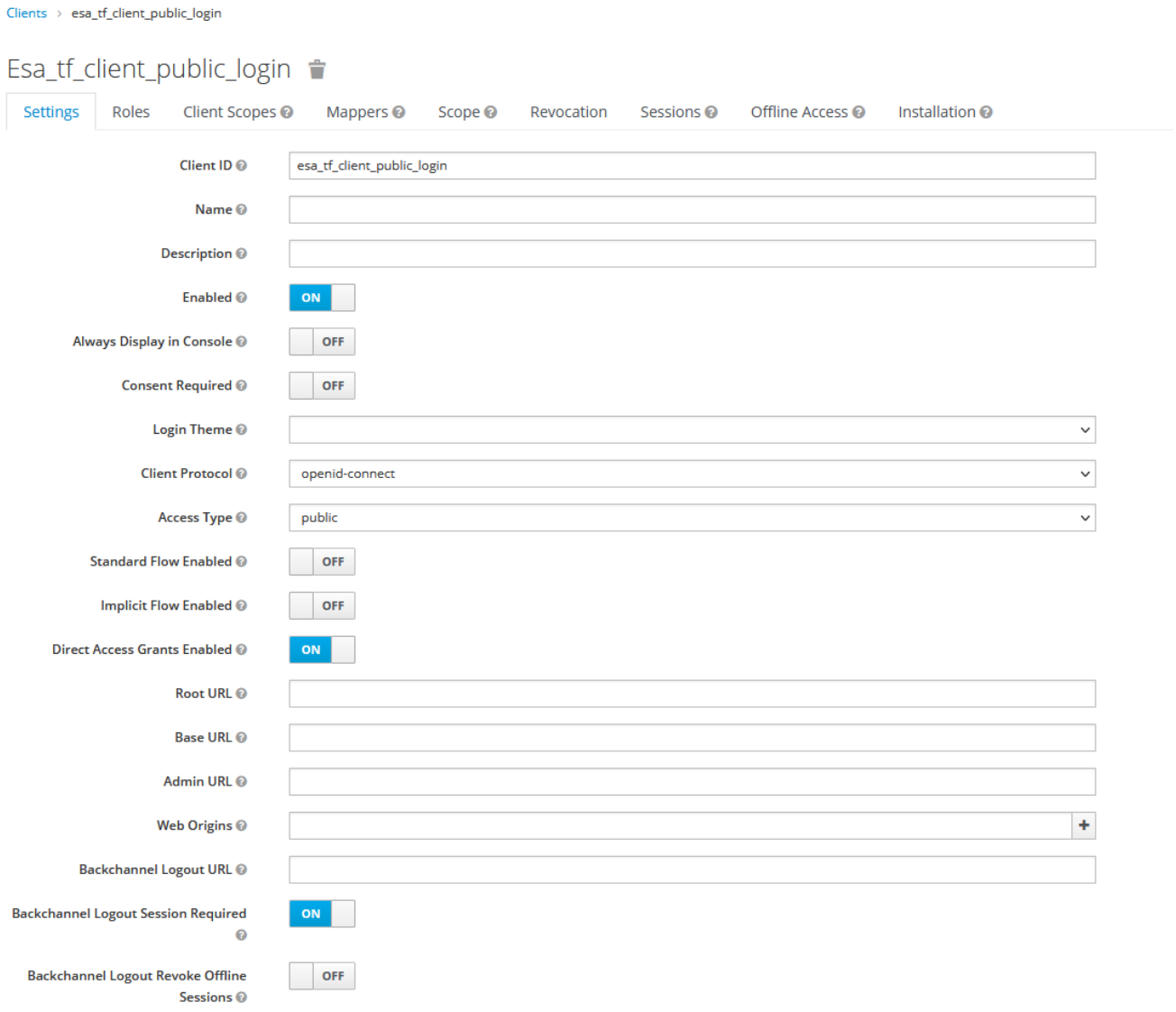

Figure 30: Transformation Framework Public Client creation

Bearer-Only Client creation

1) Access the 'Clients' section and click on 'Create'.
2) Configure:
    - Client ID = Name of the bearer-only client (e.g. esa_tf_client)
    - Client Protocol = openid-connect
3) Click on 'Save'.
4) Edit the just created client setting:
    - Access Type = bearer-only

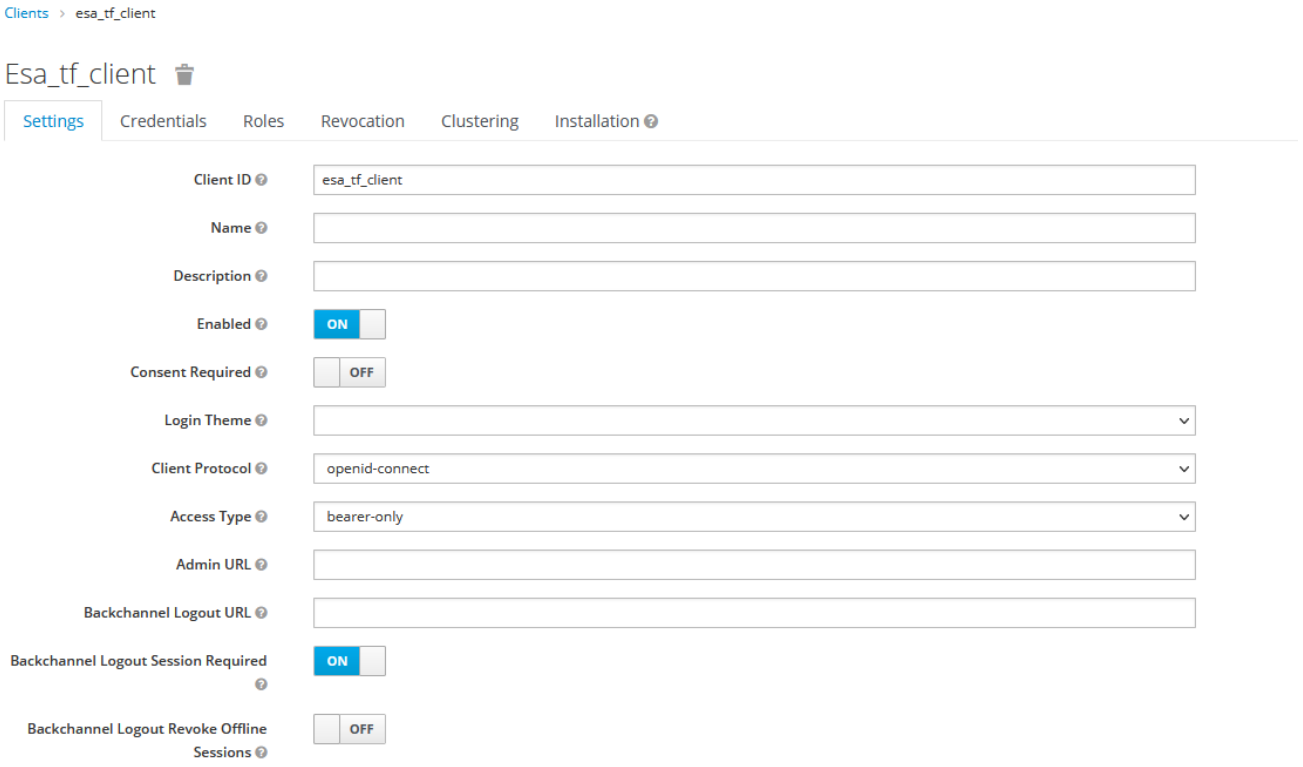

Figure 31: Transformation Framework Bearer-Only Client creation

Retrieve Secret from Bearer-Only Client

1) Access the 'Clients' section and select the Client ID associated to the "Bearer-Only" client previously created.
2) Access the 'Credentials' tab.
3) Identify the Secret present in the "Secret" field.

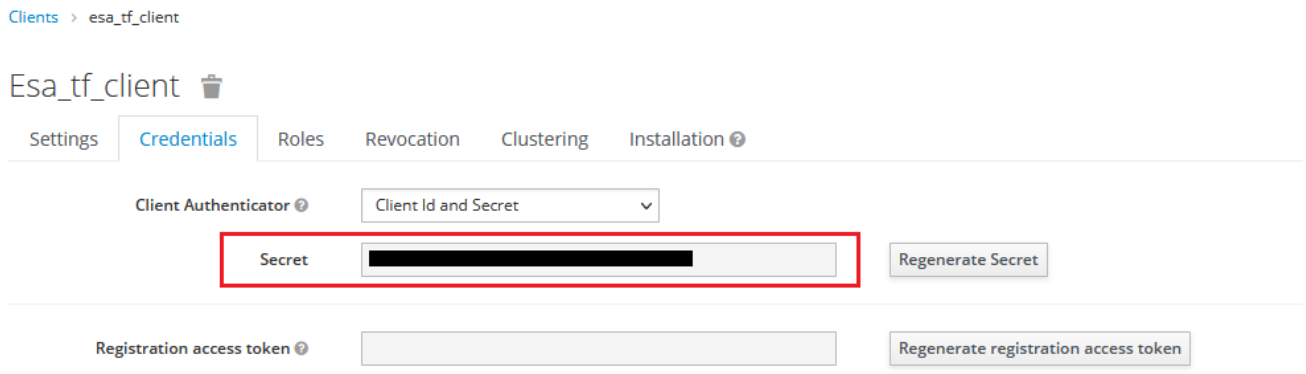

Figure 32: Transformation Framework Secret identification in Bearer-Only Client

4.1.3 User Role management

Two different User Role types are supported by the Transformation Framework:

- Standard User
- Manager User

These users shall be created in the "Bearer-Only" Client and shall be assigned to Users involved in the Transformation Framework usage.

Please refer to [RD 4] for description of allowed actions of the different users and how to configure them.

User Roles creation

1) Access the 'Clients' section and select the Client ID associated to the "Bearer-Only" client previously created.
2) Access the 'Roles' tab and click on 'Add Role'.
3) Insert a name and description. At the end of the insertion, click on 'Save'.

   Please note that:

   - The Role name can be selected according to Operational needs and shall be used in the configuration of the Transformation Framework, as reported in [RD 4].
   - The 'Description' field is optional.

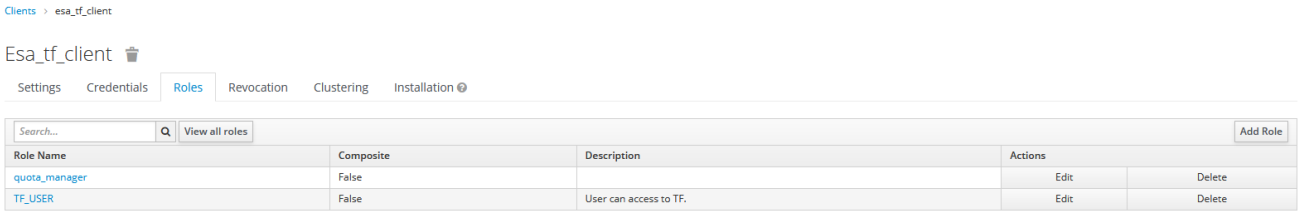

Figure 33: Transformation Framework User Roles creation

Avoid receiving User Roles from other clients

1) Access the 'Clients' section and select the Client ID associated to the "Public" client previously created.
2) Access the 'Scope' tab and put on OFF the 'Full Scope Allowed' parameter.

Refer to Figure 10.

User Role Scope Mappings setting

1) Access the 'Clients' section and select the Client ID associated to the "Public" client previously created.
2) Access the 'Scope' tab.

3) Select the Client ID associated to the "Bearer-Only" client in the 'Client Roles' dropdown menu.
4) Click on all the Roles available in the 'Available Roles' list and assign them to the client clicking the 'Add Selected' button.

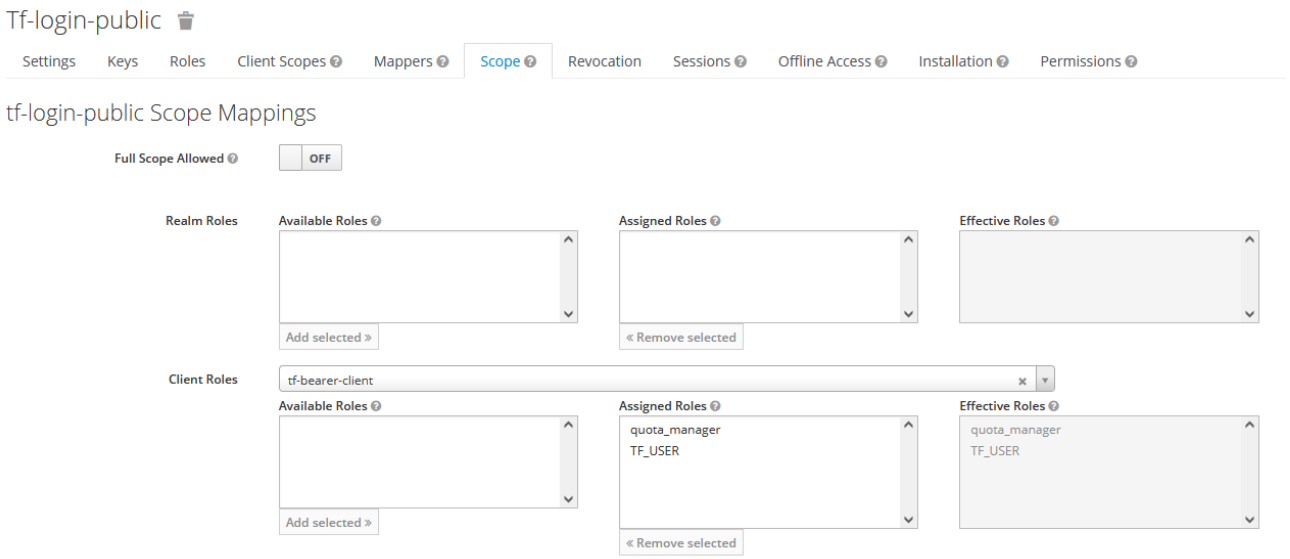

Figure 34: Transformation Framework User Role Scope Mappings setting

Assign Roles to User

1) Access the 'Users' section and search for the user to which assign the selected roles.
2) Click on the resulting User ID.
3) Access the 'Role Mappings' tab.
4) Click on the Roles available in the 'Available Roles' list of the "Realm Roles" section and assign them to the user clicking the 'Add Selected' button.
5) Select the Client ID associated to the Bearer-Only client in the 'Client Roles' dropdown menu.
6) Click on the Roles available in the 'Available Roles' list and assign them to the user clicking the 'Add Selected' button.

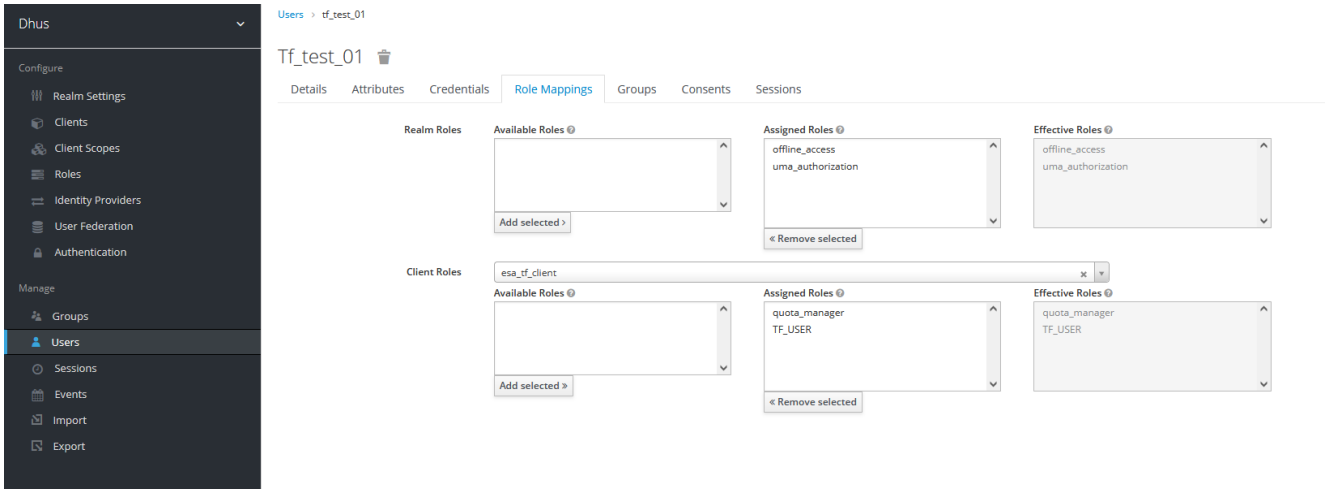

Figure 35: Transformation Framework Roles assignment

4.2 Access Token request

The Authentication flow for the Transformation Framework works as follow:

- The user must obtain an Access Token from the Public Keycloak client.
- With the obtained Access Token, the user will be able to query the Transformation Framework providing an Authorization Header.

Please note that the Transformation Framework shall be configured properly to interface with Keycloak, according to instruction in [RD 4].

Please find below a procedure to request the token and perform a request on the Transformation Framework, according to [RD 4]. Access Token could be requested using different ways (e.g curl, Postman or any other REST application). In this document we recommend the request via curl command.

1) Perform a curl query exporting a variable containing the received Access Token (AT):

   - `export AT=`curl -v -H "Content-Type: application/x-www-form-urlencoded" --data-urlencode "grant_type=password" --data-urlencode "client_id=<PUBLIC_CLIENT_ID>" --data-urlencode "username=<USERNAME>" --data-urlencode "password=<PASSWORD>" "https://<KEYCLOAK_URL>/auth/realms/<REALM_NAME>/protocol/openid-connect/token" | jq '.access_token' -r`

   Where:

   - <PUBLIC_CLIENT_ID> = Name of the public client created in Section 4.1.2 (“Public Client creation”)
   - <USERNAME> = Name of the User used to connect the Transformtion Framework

- <PASSWORD> = Password of the User used to connect the Transformtion Framework
- <KEYCLOAK_URL> = Keycloak instance URL
- <REALM_NAME> = Name of the Keycloak Realm, created in Section 4.1.1

2) Perform query towards the Transformation Framework instance providing an Authorization Header:
   - `curl "<TF_URL>/Workflows" -H "Authorization: Bearer ${AT}" | jq`
   - `curl "<TF_URL>/TransformationOrders" -H "Authorization: Bearer ${AT}" | jq`

   Where:
   - <TF_URL> = Transformation Framework URL
   - AT = Access Token variable saved by previour query at point 1)

5. GSS

In the frame of the GSS (GAEL Store Service) DHS component, the Identify and Access Management is delegated to Keycloak software.

Next Sections describe how to configure Keycloak to be compliant with actual GSS for all concerns User Management actions.

5.1 Keycloak Console Configuration

5.1.1 Realm creation

In order to create needed clients, a Realm in Keycloak is needed. The Realm can be created as follows:

- On upper left of the console, hover the Realm selection drop-down menu and click on 'Add realm'.

Note: In case a Realm is already existing (e.g. DHuS realm), the same Realm could be used to host the GSS client, sharing the same Users.

5.1.2 Clients' creation and configuration

One client is needed to integrate the GSS with Keycloak:

- A "Public" client to request and obtain an Access Token

In order to create the requested client, follow the following steps.

Public Client creation

1) Access the 'Clients' section and click on 'Create'.
2) Configure:
    - Client ID = Name of the public client (e.g. ivv-gss1-client)
    - Name = $(client_account)
    - Client Protocol = openid-connect

    Click on 'Save'.
3) Turn on "Enabled" button.
4) Edit the just created client setting:
    - Access Type = public
5) Turn to "ON" both the "Standard Flow Enable" and "Direct Access Grants Enabled" buttons.
6) Set "Web Origins" value with "+".
7) Define the "Valid Redirect URIs" links, inserting the URL pattern of the applications to be redirected after a successful login (see Figure 36). Example:

- https://<SERVICE-DOMAIN>/<PATH>/*

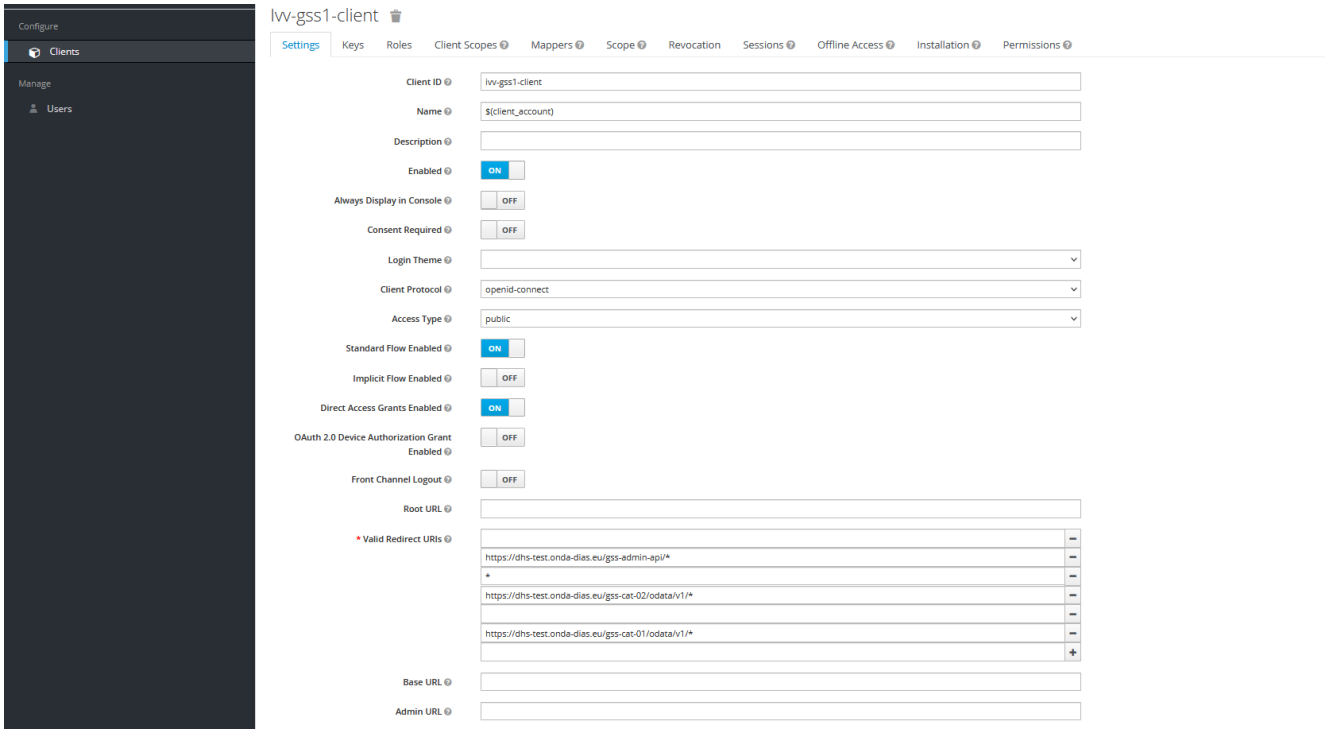

Figure 36: GSS Public Client creation

Single-Sign On (SSO) Authentication

In order to allow Users to access GSS Catalogue via Web, the following configuration shall be applied:

1) Access to 'Client Scopes' on the left panel of the selected Realm.
2) Access to 'roles'.
3) Select 'Mappers'.
4) Select 'realm roles'.
5) Click on 'Add to userinfo' to turn on the checkbox. Then click on 'Save'.
6) After this, go back to 'Mappers'.
7) Access to 'client roles'.
8) Click on 'Add to userinfo' to turn on the checkbox . Then click on 'Save'.

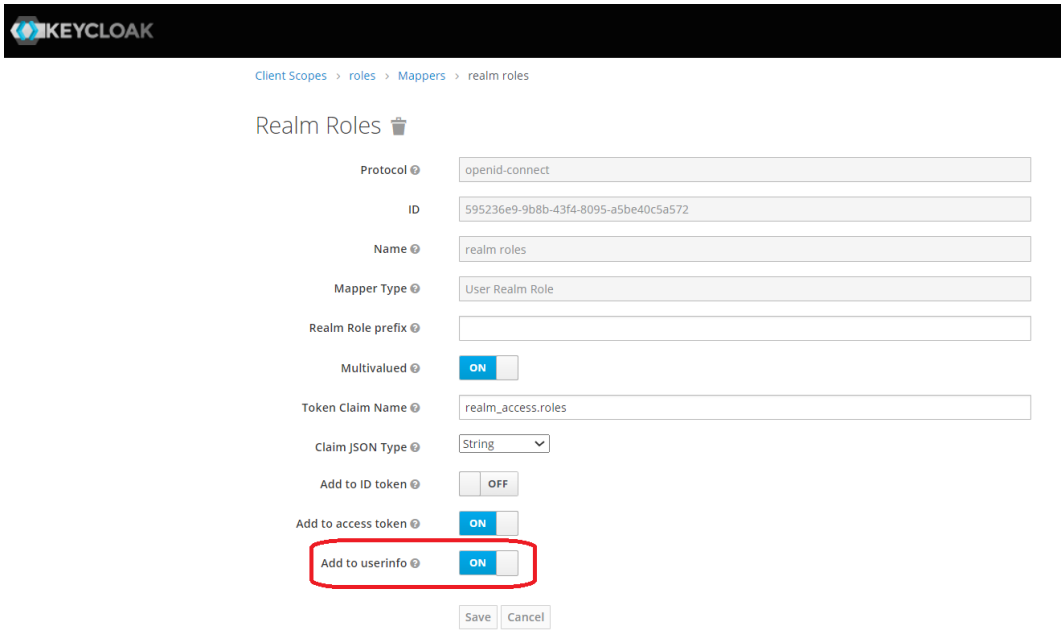

Figure 37: GSS SSO Authentication setup

5.1.3 User Role management

Two different User Role types are supported by the GSS:

- admin
- end-user

These users shall be created in the "public" Client and shall be assigned to Users involved in the GSS usage.

Please refer to [RD 5] or description of allowed actions of the different users and how to configure them.

User Roles creation

1) Access the 'Clients' section and select the Client ID associated to the "public" client previously created.
2) Access the 'Roles' tab and click on 'Add Role'.
3) Insert a name and description. At the end of the insertion, click on 'Save'.

   Please note that:

   - The user role can be selected according to Operational needs and shall be used in the configuration of the GSS, as reported in [RD 5 .
   - The 'Description' field is optional.

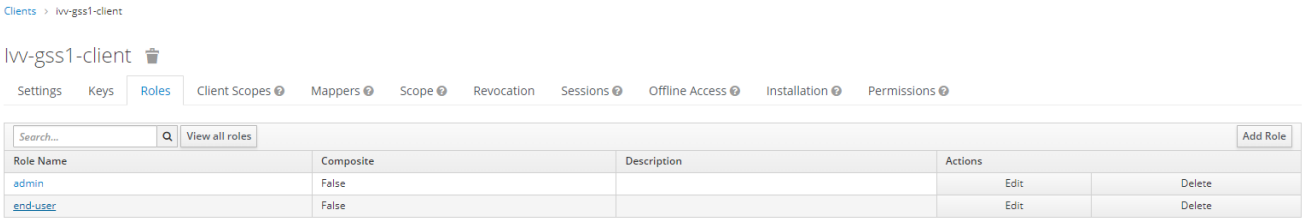

Figure 38: GSS User Roles creation

Avoid receiving User Roles from other clients

1) Access the 'Clients' section and select the Client ID associated to the "Public" client previously created.
2) Access the 'Scope' tab and put on OFF the 'Full Scope Allowed' parameter.

Refer to Figure 10.

Assign Roles to User

1) Access the 'Users' section and search for the user on which assign the selected roles.
2) Click on the resulting User ID.
3) Access the 'Role Mappings' tab.

4) Click on the Roles available in the 'Available Roles' list of the "Realm Roles" section and assign them to the user clicking the 'Add Selected' button.

5) Select the Client ID associated to the Public client in the 'Client Roles' dropdown menu.

6) Click on the Roles available in the 'Available Roles' list and assign them to the user clicking the 'Add Selected' button.

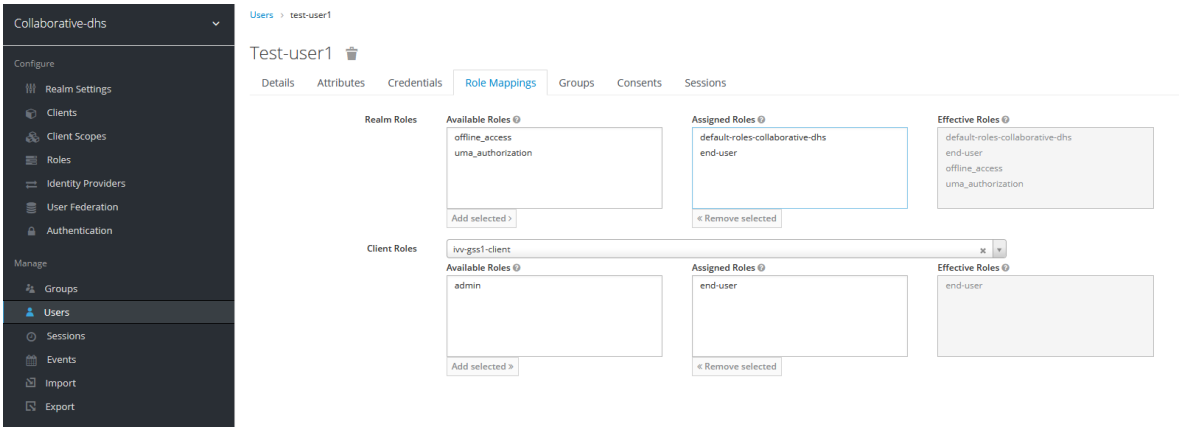

Figure 39: GSS Roles assignment

5.2 Access Token request

The Authentication flow for the GSS works as follow:

- The user must obtain an Access Token from the Public Keycloak client.
- With the obtained Access Token, the user will be able to query the GSS providing an Authorization Header.

Please note that the GSS shall be configured properly to interface with Keycloak, according to instruction in [RD 4].

Please find below a procedure to request the token and perform a request on the GSS, according to [RD 4]. Access Token could be requested using different ways (e.g curl, Postman or any other REST application). In this document we recommend the request via curl command (in case of query performed via browser to the GSS, the token request is automatically managed by the GSS itself if the authentication protocol is set with OAUTH2, according to instruction in [RD 4

1) Perform a curl query exporting a variable containing the received Access Token (AT):

- ```
  export AT=`curl -v -H "Content-Type: application/x-www-form-urlencoded" --data-urlencode "grant_type=password" --data-urlencode "client_id=<PUBLIC_CLIENT_ID>" --data-urlencode "username=<USERNAME>" --data-urlencode "password=<PASSWORD>" "https://<KEYCLOAK_URL>/auth/realms/<REALM_NAME>/protocol/openid-connect/token" | jq '.access_token' -r`
  ```

  Where:

- <PUBLIC_CLIENT_ID> = Name of the public client created in Section 5.1.2 (“Public Client creation”)
- <USERNAME> = Name of the User used to connect the GSS
- <PASSWORD> = Password of the User used to connect the GSS
- <KEYCLOAK_URL> = Keycloak instance URL
- <REALM_NAME> = Name of the Keycloak Realm, created in Section 5.1.1

2) Perform query towards the GSS instance providing an Authorization Header:

- ```
  curl "<GSS_CAT_URL_QUERY>" -H "Authorization: Bearer ${AT}" | jq
  ```

  Where:

  - <GSS_CAT_URL_QUERY> = GSS Catalogue URL query
  - AT = Access Token variable saved by previour query at point 1.