

Proceso de instalación y pruebas Keycloak en Docker

Tutorial de instalación:

<https://www.keycloak.org/getting-started/getting-started-docker>

Máquina utilizada:

- VM Ubuntu 18.04 LTS
- 2 Cores
- 2GB Ram
- 20 GB SSD

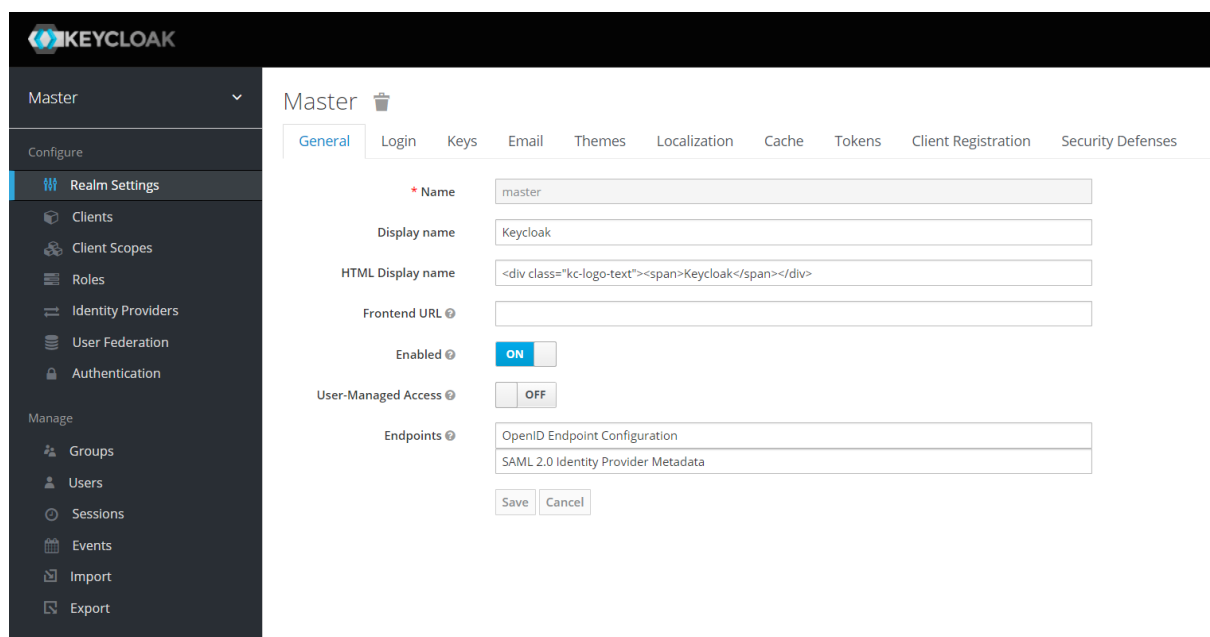
Dirección: <https://keycloak.diinf.tk/>

Credenciales para consola de administración: admin - Operaciones1*

Se modificó un poco el comando de **docker run** de la documentación para poder utilizar https:

```
docker run -d --name keycloak_app -p 443:8443 -e KEYCLOAK_USER=admin -e  
KEYCLOAK_PASSWORD=Operaciones1* quay.io/keycloak/keycloak:14.0.0
```

En caso de no ser ingresados, Keycloak genera sus propios certificados autofirmados.



Vista principal de administración

Posterior a esto, se siguieron todos los pasos de la documentación, estableciendo el **REALM**, usuario de prueba y cliente de prueba.

The screenshot displays the Keycloak administration console. On the left is a dark sidebar with a menu. The 'Operaciones' realm is selected, and the 'General' tab is active. The main content area shows configuration fields for the realm. The 'Name' field is set to 'Operaciones'. The 'Display name' is also 'Operaciones'. The 'HTML Display name' is empty. The 'Frontend URL' is empty. The 'Enabled' toggle is turned 'ON'. The 'User-Managed Access' toggle is turned 'OFF'. The 'Endpoints' section lists 'OpenID Endpoint Configuration' and 'SAML 2.0 Identity Provider Metadata'. At the bottom are 'Save' and 'Cancel' buttons.

Field	Value
Name	Operaciones
Display name	Operaciones
HTML Display name	
Frontend URL	
Enabled	ON
User-Managed Access	OFF
Endpoints	OpenID Endpoint Configuration, SAML 2.0 Identity Provider Metadata

Vista principal de administración de realm Operaciones

The screenshot shows the Keycloak administration console with the 'Users' section selected in the sidebar. The 'Test.user' user is selected, and the 'Details' tab is active. The main content area displays user information. The 'ID' is '030a05f0-68be-49aa-8599-a904b149340c'. The 'Created At' timestamp is '5/28/21 4:33:34 PM'. The 'Username' is 'test.user'. The 'Email' field is empty. The 'First Name' is 'Test' and the 'Last Name' is 'User'. The 'User Enabled' and 'Email Verified' toggles are both turned 'ON'. The 'Required User Actions' dropdown is set to 'Select an action...'. The 'Impersonate user' button is labeled 'Impersonate'. At the bottom are 'Save' and 'Cancel' buttons.

Field	Value
ID	030a05f0-68be-49aa-8599-a904b149340c
Created At	5/28/21 4:33:34 PM
Username	test.user
Email	
First Name	Test
Last Name	User
User Enabled	ON
Email Verified	ON
Required User Actions	Select an action...
Impersonate user	Impersonate

Creación usuario de prueba

The screenshot shows the Keycloak Admin Console interface. On the left is a sidebar with navigation menus: 'Operaciones' (with a dropdown), 'Configure' (containing 'Realm Settings', 'Clients', 'Client Scopes', 'Roles', 'Identity Providers', 'User Federation', and 'Authentication'), and 'Manage' (containing 'Groups', 'Users', 'Sessions', 'Events', 'Import', and 'Export'). The 'Clients' menu item is selected. The main content area is titled 'Clients > test-client' and 'Test-client'. Below the title are tabs for 'Settings', 'Roles', 'Client Scopes', 'Mappers', 'Scope', 'Revocation', 'Sessions', 'Offline Access', and 'Installation'. The 'Settings' tab is active, displaying various configuration fields: 'Client ID' (test-client), 'Name' (empty), 'Description' (empty), 'Enabled' (OFF), 'Always Display in Console' (ON), 'Consent Required' (OFF), 'Login Theme' (dropdown), 'Client Protocol' (openid-connect), 'Access Type' (public), and 'Standard Flow Enabled' (ON).

Registro de aplicación de prueba

The screenshot shows a configuration page with a red geometric border. It contains three input fields: 'Keycloak URL' with the value 'https://keycloak.operaciones.tk/auth', 'Realm' with the value 'Operaciones', and 'Client' with the value 'test-client'. Below these fields is a blue 'Save' button.

Configuración en <https://www.keycloak.org/app/>

The screenshot shows a user interface with a red geometric border. At the top, there are two buttons: a blue 'Sign out' button and a white 'Clear config' button. Below these buttons, a white box contains the text 'Hello, Test User'.

DRespuesta post autenticación

Levantamiento de contenedor con TLS/SSL

Información obtenida desde:

1. <https://stackoverflow.com/questions/52674979/keycloak-ssl-setup-using-docker-image>
2. <https://stackoverflow.com/questions/58137934/keycloak-from-docker-letsencrypt-cert-and-err-ssl-version-or-cipher-mismatch>

Primero es necesario generar el certificado, esto se realizó mediante **Letsencrypt**, particularmente mediante **Cerbot** mediante el comando:

```
sudo certbot certonly --standalone
```

Cuando consulte por el dominio, en este caso se utilizó **keycloak.diinf.tk**

El siguiente paso es mover y cambiar el nombre de los archivos generados, se utilizó la ruta “**/certs**”.

```
sudo cp /etc/letsencrypt/live/keycloak.diinf.tk/fullchain.pem certs/tls.crt
sudo cp /etc/letsencrypt/live/keycloak.diinf.tk/privkey.pem certs/tls.key
```

```
sudo chmod 755 certs/
sudo chmod 655 certs/*
```

Finalmente, se puede levantar el contenedor recordando montar el volumen para incluir los certificados dentro del contenedor, se puede hacer mediante el siguiente comando:

```
docker run -d --name keycloak_app -p 443:8443 \
-v /home/robert/Keycloak/certs:/etc/x509/https \
-e KEYCLOAK_USER=admin \
-e KEYCLOAK_PASSWORD=Operaciones1* \
quay.io/keycloak/keycloak:14.0.0
```

Finalmente, se puede revisar el certificado en el navegador:



User Storage Federation con SAMBA AD

- 1) El servidor de SAMBA fue levantado con docker-compose y con el hostname samba.diinf.tk
- 2) El servidor de Keycloak fue levantado sin sus certificados tls, hay que revisar cómo modificar dinámicamente standalone-ha.xml para poder agregarlos.

Primero es necesario obtener el certificado del servidor de SAMBA en la máquina que tiene a Keycloak, para esto se puede utilizar el siguiente comando:

```
openssl s_client -connect SERVER-FQDN:636 </dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > cert.pem
```

Luego hay que ingresar este certificado a un Keystore para que Keycloak pueda confiar en que el servidor al que se está consultado es conocido, el archivo **truststore.jks** se crea con el siguiente comando (luego de crearlo, usar chmod para modificar los permisos y que Docker pueda leerlo posteriormente):

```
keytool -import -trustcacerts -file cert.pem -alias SERVER-FQDN -keystore truststore.jks
```

```
sudo chmod 655 truststore.jks
```

También es necesario crear/modificar el archivo **standalone-ha.xml** y agregar la siguiente sección SPI (el value de password “changeit” corresponde a la ingresada al momento de crear el archivo **truststore.jks**):

```
...
<spi name="truststore">
  <provider name="file" enabled="true">
    <properties>
      <property name="file" value="/opt/jboss/keycloak/standalone/configuration/truststore.jks"/>
      <property name="password" value="changeit"/>
      <property name="hostname-verification-policy" value="WILDCARD"/>
      <property name="enabled" value="true"/>
    </properties>
  </provider>
</spi>
...
```

Finalmente, iniciar el servidor de Keycloak mediante docker:

```
docker run -d --name keycloak_app -p 443:8443 \
  -v truststore.jks:/opt/jboss/keycloak/standalone/configuration/truststore.jks \
  -v standalone-ha.xml:/opt/jboss/keycloak/standalone/configuration/standalone-ha.xml \
  -e KEYCLOAK_USER=admin \
  -e KEYCLOAK_PASSWORD=Operaciones1* \
  quay.io/keycloak/keycloak:14.0.0
```

Detalles de levantamiento:

- 1) El archivo **standalone-ha.xml** lo conseguí previamente levantando un contenedor sin configuraciones y copiando con el siguiente comando:

```
docker cp c_name:/opt/jboss/keycloak/standalone/configuration/standalone-ha.xml standalone-ha.xml
```

Luego este le agregué la parte del SPI del Truststore.

- 2) Si se levanta el contenedor con certificados tls, el script de Keycloak realiza una configuración en **standalone-ha.xml** (probablemente alguna llave generada aleatoriamente es guardada), por lo que de momento no es posible utilizar estos certificados.

Configuración User Storage Federation

Primero es necesario crear un nuevo realm dentro de Keycloak (no usar el master realm por recomendación de la documentación). Luego de seleccionar este real, ir a la sección de User Federation y agregar un provider de tipo ldap:

Operaciones

Configure

Manage

Realm Settings

Clients

Client Scopes

Roles

Identity

Providers

User Federation

Authentication

Groups

Users

Sessions

Events

Import

Export

User Federation > Add user storage provider

Add user federation provider

Required Settings

Enabled

ON

Console Display Name

Samba

Priority

0

Import Users

ON

Edit Mode

WRITABLE

Sync Registrations

ON

* Vendor

Active Directory

* Username LDAP attribute

sAMAccountName

* RDN LDAP attribute

cn

* UUID LDAP attribute

objectGUID

* User Object Classes

person, organizationalPerson, user

* Connection URL

ldaps://samba.operaciones.tk:636

Test connection

* Users DN

CN=Users,DC=operaciones,DC=tk

Custom User LDAP Filter

LDAP Filter

Search Scope

One Level

* Bind Type

simple

* Bind DN

CN=Administrator,CN=Users,DC=operaciones,DC=tk

* Bind Credential

Operaciones1*

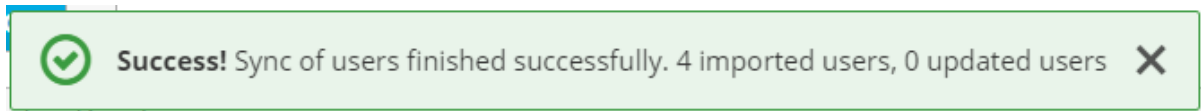
Test authentication

Si los dos botones **Test connection** y **Test authentication** entregan success, guardar la configuración.

Al final de las configuraciones aparecerán las siguientes opciones



Al presionar el botón **Synchronize all users** se hará una copia de todos los usuarios de SAMBA hacia Keycloak.



Los cuales pueden ser luego verificados en la pestaña de usuarios.

Users

Lookup

<input type="text" value="Search..."/>	<input type="button" value="Q"/>	<input type="button" value="View all users"/>	<input type="button" value="Unlock users"/>					<input type="button" value="Add user"/>
ID	Username	Email	Last Name	First Name	Actions			
bd38149b-c694-494...	administrator				Edit	Impersonate	Delete	
5649af3c-db2f-4183-...	guest				Edit	Impersonate	Delete	
7f0aea62-adda-4060...	krbtgt				Edit	Impersonate	Delete	
533938f6-0332-43d2...	roberto.lillo	roberto.lillo@usach.cl	Lillo	Roberto	Edit	Impersonate	Delete	

También se puede ver que provienen de SAMBA gracias al parámetro Federation Link.

Users > roberto.lillo

Roberto.lillo

Details	Attributes	Credentials	Role Mappings	Groups	Consents	Sessions
ID	533938f6-0332-43d2-bb02-c98134c57ea4					
Created At	7/3/21 5:37:35 PM					
Username	roberto.lillo					
Email	roberto.lillo@usach.cl					
First Name	Roberto					
Last Name	Lillo					
User Enabled	<input checked="" type="checkbox"/>					
Federation Link	Samba					
Email Verified	<input type="checkbox"/>					
Required User Actions	Select an action...					
Impersonate user	<input type="button" value="Impersonate"/>					
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>					

Pruebas realizadas

1) Cambiar el apellido de un usuario

Primero hice un ldapsearch para buscar el usuario “roberto.lillo” y tiene el siguiente sn (surname).

```
# Roberto Lillo, Users, operaciones.tk
dn: CN=Roberto Lillo,CN=Users,DC=operaciones,DC=tk
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Roberto Lillo
sn: Lillo
givenName: Roberto
```

Luego, dentro de la interfaz de Keycloak, cambié el apellido por “Toloza” y guardé el cambio.

Username	<input type="text" value="roberto.lillo"/>
Email	<input type="text" value="roberto.lillo@usach.cl"/>
First Name	<input type="text" value="Roberto"/>
Last Name	<input type="text" value="Toloza"/>
User Enabled ?	<input checked="" type="checkbox"/> ON

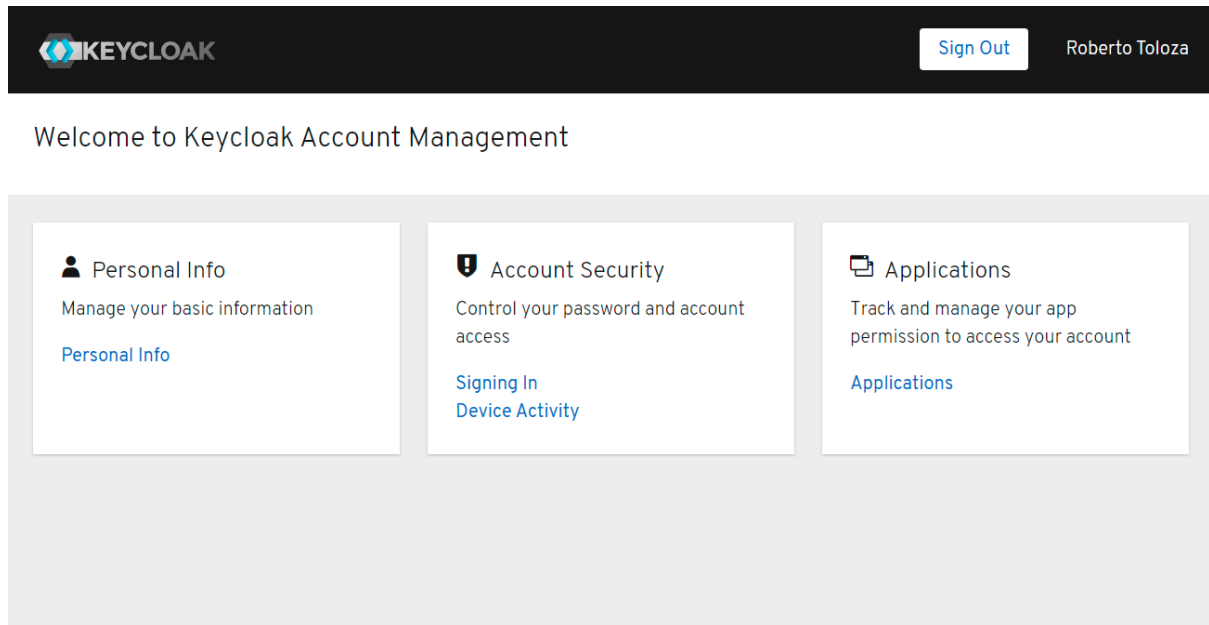
Para finalizar hice un nuevo ldapsearch buscando al mismo usuario.

```
mail: roberto.lillo@usach.cl
userAccountControl: 512
memberOf: CN=Alumno,CN=Users,DC=operaciones,DC=tk
pwdLastSet: -1
sn: Toloza
whenChanged: 20210703214544.0Z
uSNChanged: 3777
distinguishedName: CN=Roberto Lillo,CN=Users,DC=operaciones,DC=tk
```


También se puede apreciar que cambió el timestamp del atributo “whenChanged”.

2) Cambiar la contraseña del usuario desde la consola de administración

Primero ingresé a la consola de usuarios con el usuario “roberto.lillo” y la contraseña “Operaciones1*”.



Luego, cerré sesión y cambié la contraseña del usuario en la ventana de administración.

Roberto.lillo 

[Details](#) [Attributes](#) [Credentials](#) [Role Mappings](#) [Groups](#) [Consents](#) [Sessions](#)


Supported User Storage Credential Types


Type	Provided By
password	Samba


Manage Credentials

Position	Type	User Label	Data	Actions
----------	------	------------	------	---------

Reset Password

Password 

Password Confirmation 

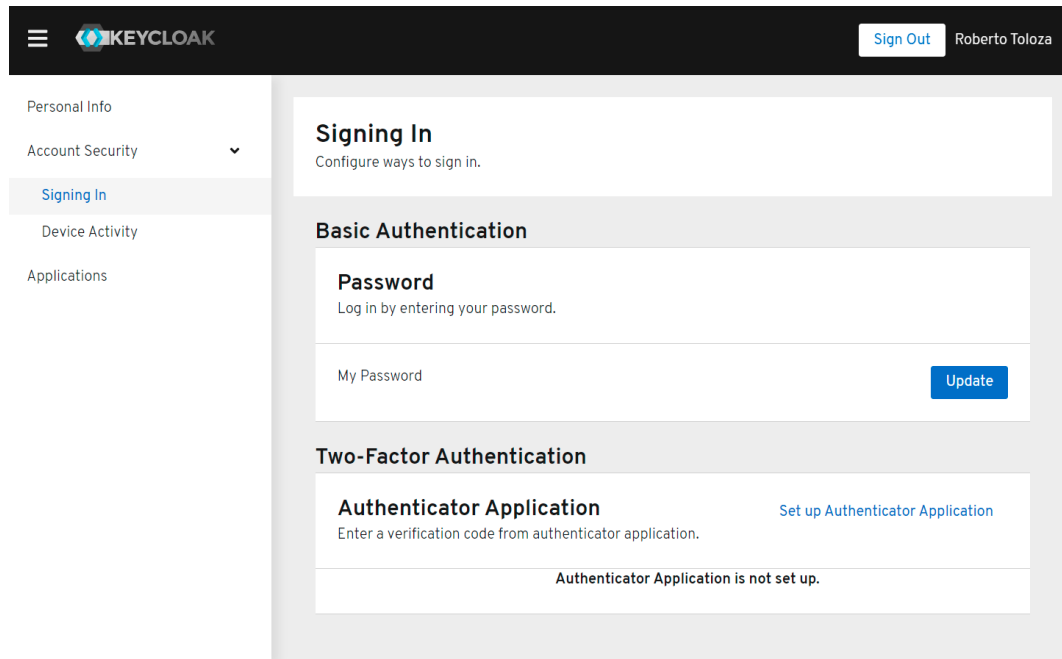
Temporary  ☐ OFF

Finalmente, es posible ingresar nuevamente a la consola de usuarios con la nueva contraseña "PasswordTest1*". Incluso mediante ldapsearch es posible ver que el timestamp del atributo "**whenChanged**" se actualizó.

```
lastLogon: 132698228994508000
lastLogonTimestamp: 132698228994508000
whenChanged: 20210703215852.0Z ←
pwdLastSet: -1
uSNChanged: 3780
distinguishedName: CN=Roberto Lillo,CN=Users,DC=operaciones,DC=tk
```

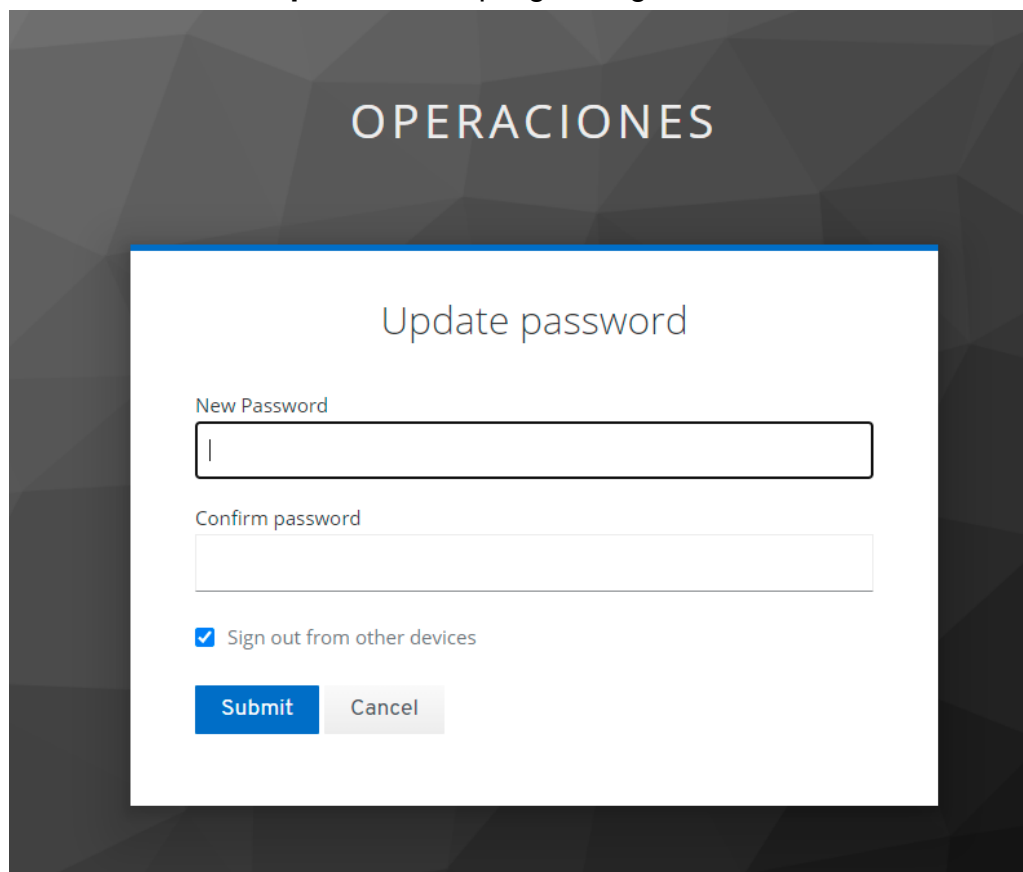
3) Cambiar la contraseña del usuario desde la consola del usuario

Primero ingresé a la consola del usuario y luego a la sección “Account Security” seguido de “Signing In”.



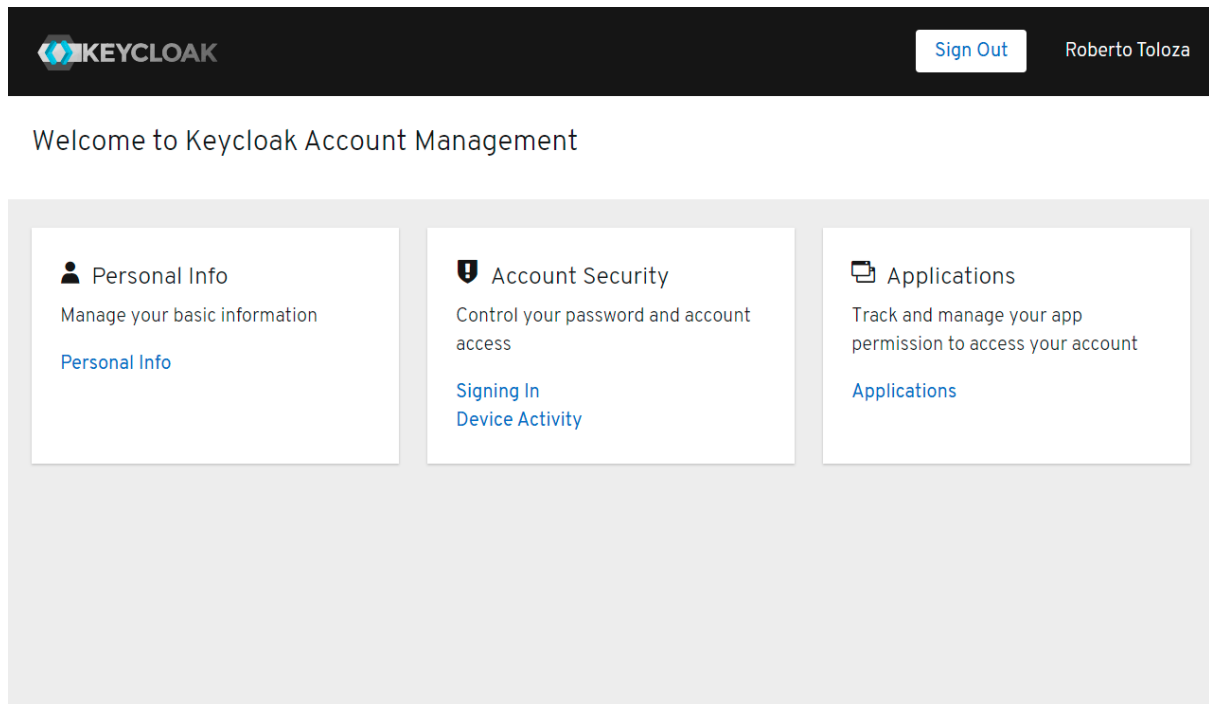
The screenshot shows the Keycloak user console interface. At the top, there is a navigation bar with the Keycloak logo and a 'Sign Out' button next to the user name 'Roberto Toloza'. On the left, a sidebar menu lists 'Personal Info', 'Account Security' (expanded), 'Signing In' (selected), 'Device Activity', and 'Applications'. The main content area is titled 'Signing In' with the subtitle 'Configure ways to sign in.' It contains two sections: 'Basic Authentication' and 'Two-Factor Authentication'. Under 'Basic Authentication', there is a 'Password' section with the instruction 'Log in by entering your password.' Below this is a 'My Password' field and an 'Update' button. Under 'Two-Factor Authentication', there is an 'Authenticator Application' section with the instruction 'Enter a verification code from authenticator application.' and a link 'Set up Authenticator Application'. A status message at the bottom of this section says 'Authenticator Application is not set up.'

Al presionar en el botón **Update** se despliega la siguiente ventana.



The screenshot shows a modal window titled 'Update password' over a dark background with the word 'OPERACIONES' at the top. The modal contains two input fields: 'New Password' and 'Confirm password'. Below these fields is a checkbox labeled 'Sign out from other devices' which is checked. At the bottom of the modal are two buttons: 'Submit' and 'Cancel'.

Se vuelve a ingresar la contraseña “Operaciones1*” y se presiona el botón **Submit**. Esto lleva nuevamente a la consola del usuario. Si se cierra la sesión en esta y se vuelve a iniciar sesión con la contraseña nueva, se puede ingresar sin problemas.




Nuevamente, utilizando ldapsearch se puede verificar otra vez que el timestamp de “**whenChanged**” también cambió.

```
lastLogon: 132698228994508000
lastLogonTimestamp: 132698228994508000
whenChanged: 20210703220547.0Z ←
pwdLastSet: -1
uSNChanged: 3782
distinguishedName: CN=Roberto Lillo,CN=Users,DC=operaciones,DC=tk
```

4) Crear un usuario en Keycloak

Dentro de la pestaña de usuarios se creó un nuevo usuario.

[Users](#) > francisco.guajardo.v

Francisco.guajardo.v 

Details	Attributes	Credentials	Role Mappings	Groups	Consents	Sessions
ID	80f40969-f0b9-4325-8675-9b994160a95c					
Created At	7/3/21 6:19:27 PM					
Username	francisco.guajardo.v					
Email	francisco.guajardo.v@usach.cl					
First Name	Francisco					
Last Name	Guajardo					
User Enabled ?	<input checked="" type="checkbox"/> ON					
Federation Link ?	Samba					
Email Verified ?	<input checked="" type="checkbox"/> ON					
Required User Actions ?	<input checked="" type="checkbox"/> Update Password					
Impersonate user ?	<input type="button" value="Impersonate"/>					
<input type="button" value="Save"/> <input type="button" value="Cancel"/>						

Se puede apreciar que automáticamente se activó el User Storage Federation gracias al atributo Federation Link. De forma automática también se activó la acción de requerir que el usuario actualice su password cuando ingrese por primera vez.

Usando ldapsearch se puede buscar al usuario también en el servidor de SAMBA.

```
# francisco.guajardo.v, Users, operaciones.tk
dn: CN=francisco.guajardo.v,CN=Users,DC=operaciones,DC=tk
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: francisco.guajardo.v
instanceType: 4
whenCreated: 20210703221927.0Z
whenChanged: 20210703221927.0Z
```

Al intentar iniciar sesión en la consola de usuarios, fue necesario cambiar la contraseña para poder ingresar (detalle con cuál es la contraseña por defecto).