

# OPENID Keycloak Anbindung Proxmox

Du möchtest dich gerne für unsere Hilfe erkenntlich zeigen 😊. Gerne. Wir bedanken uns bei dir für deine Spende! ☺

[Spenden](#)

Zum frei verfügbaren [Apt-Repository](#)



GITLAB:

Die folgende Dokumentation zeigt die Keycloakanbindung von Proxmox inkl. Login berechtigten Gruppen. Als Backend wird [LDAP von UCS \(Univention\)](#) verwendet. Für das ganze Vorhaben wird eine ähnlich funktionierende Umgebung voraus gesetzt.

Verwendete Systeme/Software:

- Proxmox 8.1.4
- UCS 5.0-6 errata993
- Keycloak installiert am Primary Directory Node (ohne verteilter Datenbank) 23.0.7

Proxmoxclusternodes:

- **pve01.tux.lan**
- **pve02.tux.lan**
- **pve03.tux.lan**

Primary Directory Node: **dc1.tux.lan**

## OpenID Client in Keycloak hinzufügen

Unter dem Realm „ucs“ wird ein neuer Client names „proxmox-cluster01“ hinzugefügt. Die Basiseinrichtung erfolgt in 3 Schritten:

1. Erstellen eines neuen Clients

The screenshot shows the Keycloak administration interface. The left sidebar has a dropdown menu set to 'ucs'. The 'Clients' tab is selected. The main content area is titled 'Clients' with a sub-instruction: 'Clients are applications and services that can request authentication of a user.' Below this are three tabs: 'Clients list' (selected), 'Initial access token', and 'Client registration'. At the bottom of the main area is a search bar and a prominent blue 'Create client' button. The entire interface is dark-themed.

## 2. Setzen des „Client type“ und der „Client-ID“

Clients > Create client

### Create client

Clients are applications and services that can request authentication of a user.

1 General settings      2 Capability config      3 Login settings

Client type	OpenID Connect
Client-ID *	proxmox-cluster01
Name	
Beschreibung	
Always display in UI	<input checked="" type="checkbox"/> Off

Next Back Abbrechen

Clients > Create client

### Create client

Clients are applications and services that can request authentication of a user.

1 General settings      2 Capability config      3 Login settings

Client authentication	<input checked="" type="checkbox"/> Off	
Authorization	<input checked="" type="checkbox"/> Off	
Authentication flow	<input checked="" type="checkbox"/> Standard flow <input type="checkbox"/> Implicit flow <input type="checkbox"/> OAuth 2.0 Device Authorization Grant <input type="checkbox"/> OIDC CIBA Grant	<input checked="" type="checkbox"/> Direct access grants <input type="checkbox"/> Service accounts roles

Next Back Abbrechen

## 3. Setzen der „Valid redirect URI's

Root URL:

Home URL:

Valid redirect URIs:

- https://pve01.tux.lan:8006
- https://pve02.tux.lan:8006
- https://pve03.tux.lan:8006

Add valid redirect URIs

Valid post logout redirect URIs:   
Add valid post logout redirect URIs

Web Origins:   
Add web origins

**Speichern** Back Abbrechen

Für unser späteres Vorhaben „nur bestimmte Gruppen zu zulassen“, müssen nach dem „Speichern“ noch zwei weitere Optionen unter „**Einstellungen**“ aktiviert werden. Dies schaltet weitere Funktionen frei.

**Capability config**

Client authentication  On

Authorization  On

Authentication flow	<input checked="" type="checkbox"/> Standard flow	<input checked="" type="checkbox"/> Direct access grants
	<input type="checkbox"/> Implicit flow	<input checked="" type="checkbox"/> Service accounts roles
	<input type="checkbox"/> OAuth 2.0 Device Authorization Grant	
	<input type="checkbox"/> OIDC CIBA Grant	

Damit wäre die Basiseinrichtung abgeschlossen.

## Konfiguration OpenID auf Proxmox

Hier bedient man sich am besten der CMD. Bevor man dies tut muss man sich aber noch das „**Client Secret**“ kopieren.

The screenshot shows the Keycloak admin UI with the 'Clients' tab selected. Under the 'Clients' tab, there is a list of clients. One client is selected, named 'proxmox-cluster01'. The 'openid-connect' tab is active. On the right, there are several configuration sections: 'Client Authenticator' (set to 'Client Id and Secret'), 'Client Secret' (a masked password field with a 'Regenerate' button), and 'Registration access token' (another masked field with a 'Regenerate' button). A red arrow points to the 'Regenerate' button next to the Client Secret.

Danach wird folgender Befehl auf der Rootshell von Proxmox abgesetzt:

```
pveum realm add tux.lan-SSO --type openid --issuer-url  
https://ucs-sso-ng.tux.lan/realms/ucs --client-id proxmox-cluster01 --  
client-key XXXXX --username-claim username
```

–autocreate wäre optional. Damit werden Benutzer beim Ersten Login automatisch angelegt. Ab dem Zeitpunkt ist der Login mittels SSO/SAML möglich. Man hat aber noch keine Rechte.

Berechtigungen müssen manuell im Proxmox Webinterface für den/die Benutzer hinzugefügt werden. Berechtigungen werden „on the fly“ übernommen.

Die Empfehlung ist hier eine Gruppe im Proxmox Webinterface zu erstellen und den Benutzer dort einfach hinzuzufügen.

Datacenter → Permissions → Groups Gruppe anlegen, z.B. „admin“

Datacenter → Permissions → Users Gewünschten Benutzer bearbeiten und die Gruppe „admin“ zuweisen.

## Einschränkung auf Gruppen

Um überhaupt zu den LDAP-Gruppen zu kommen, muss ein „group-ldap-mapper“ hinzugefügt werden. Hierzu wechselt man im Menü von Keycloak auf „User federation“ und bearbeitet den „ldap-provider“. Im TAB Mappers, fügt man nun den „group-ldap-mapper“ hinzu.

User federation > Einstellungen

## LDAP

Einstellungen	Mappers
<input type="text" value="Search for mapper"/> →	<button>Add mapper</button>
Name	Typ
creation date	user-attribute-ldap-mapper
displayName	user-attribute-ldap-mapper
email	user-attribute-ldap-mapper
entryUUID	user-attribute-ldap-mapper
first name	user-attribute-ldap-mapper
group-ldap-mapper	group-ldap-mapper
last name	user-attribute-ldap-mapper
modify date	user-attribute-ldap-mapper
second-mail	user-attribute-ldap-mapper
uid	user-attribute-ldap-mapper
Univention ldap mapper	univention-ldap-mapper
username	user-attribute-ldap-mapper

Der Inhalt wurde auf einen Default UCS-LDAP angepasst.

Attributbeschreibung	Attributname	Info
ID	auto generiert	
Name	group-ldap-mapper	
Mapper type	group-ldap-mapper	
LDAP Groups DN	cn=tux-groups,cn=groups,dc=tux,dc=lan	Beispiel
Group Name LDAP Attribute	cn	
Group Object Classes	posixGroup	
Preserve Group Inheritance	OFF	
Ignore Missing Groups	OFF	
Membership LDAP Attribute	memberUid	
Membership Attribute Type	UID	
Membership User LDAP Attribute	uid	
LDAP Filter	(&(uid=%s)(memberof=cn=proximoxi,cn=tux-groups,cn=groups,dc=tux,dc=lan))	Kann verwendet werden um noch granularer zu werden.
Mode	READ_ONLY	
User Groups Retrieve Strategy	LOAD_GROUPS_BY_MEMBER_ATTRIBUTE	
Member-Of LDAP Attribute	memberOf	

Attributbeschreibung	Attributname	Info
Mapped Group Attributes		
Drop non-existing groups during sync	OFF	
Groups Path	/	Dies zu Ändern macht bei vielen Gruppen vielleicht Sinn.

Danach „**Speichern**“, nochmal einsteigen und rechts oben auf „**Aktion → Sync LDAP groups to Keycloak**“ anklicken. Damit sollte eine grüne Infomeldung aufpoppen wo die gesyncnten Gruppen angezeigt werden. Damit sind unter „**Gruppen**“ nun auch alle Gruppen und Groupmembers in Keycloak ersichtlich.

## Userimport und automatischer Sync in Echtzeit (optional)

Dieser Schritt muss nicht durchgeführt werden. Keycloak schaut auch jedes mal gerne am LDAP Live nach welche Benutzer es gibt. Aus Performancegründe macht es bei größeren Installationen Sinn die Benutzer direkt in die lokal MariaDB zu syncen. Hier zu bearbeitet man wieder den „**ldap-provider**“ und aktiviert bei „**Synchronization settings**“ das Flag bei **Import users**. Danach einmal abspeichern.

### Synchronization settings

Import users <span style="color: #007bff;">?</span>	<input checked="" type="checkbox"/> On
Sync Registrations <span style="color: #007bff;">?</span>	<input type="checkbox"/> Off
Batch size <span style="color: #007bff;">?</span>	1000
Periodic full sync <span style="color: #007bff;">?</span>	<input type="checkbox"/> Off
Periodic changed users sync <span style="color: #007bff;">?</span>	<input type="checkbox"/> Off

Jetzt hat man in der rechten oberen Ecke unter „**Aktion**“ eine neue freigeschaltete Funktion: „**Sync all users**“. Hier sollte wiedermals eine grüne Infomeldung auftauchen wo die gesyncnten Benutzer angezeigt werden. Nach dieser Aktion möchte man auch noch den Livesync der User und Gruppenmitgliedschaften aktivieren. Hierzu noch das folgende Flag auf „On“ schalten.

### Synchronization settings

Import users  On

Sync Registrations  Off

Batch size

Periodic full sync  Off

Periodic changed users sync  On

Changed users sync period

Beim Speichern der Einstellungen deaktiviert sich dieses Flag wieder. ...scheint aber zu funktionieren, weil die richtigen Info's in Keycloak angezeigt werden.

## Einrichtung der Authorization im proxmox-cluster01 Client

Als erstes muss im Client die „Default Policy gelöscht werden“.

Name	Type	Dependent permission	Beschreibung
> Default Policy	Jt	Default Permission	A policy that grants access only for users within this realm

Nun fügen wir eine neue Group-Policy hinzu.

Name	Beschreibung
Client	Define conditions for your permissions where a set of one or more clients is permitted to access an object.
Client Scope	Define conditions for your permissions where a set of one or more client scopes is permitted to access an object.
Group	Define conditions for your permissions where a set of one or more groups (and their hierarchies) is permitted to access an object.
Regex	Define regex conditions for your permissions.
Role	Define conditions for your permissions where a set of one or more roles is permitted to access an object.
Time	Define time conditions for your permissions.
User	Define conditions for your permissions where a set of one or more users is permitted to access an object.

## Create group policy

Name \* ⓘ proxmox-zugang01

Beschreibung ⓘ Gruppeneinschraenkung fuer den Clusterzugang

Groups claim ⓘ

Gruppen \* ⓘ [Add groups](#)

Gruppen	Extend to children
/testgruppebla	<input type="checkbox"/> ⚙
/gitlab	<input type="checkbox"/> ⚙

Logic ⓘ  Positive  Negative

[Speichern](#) [Abbrechen](#)

 Die folgenden zwei Gruppen „testgruppebla“ und „gitlab“ sollen also berechtigt sein sich auf dem Cluster einloggen zu können.

Im nächsten Schritt fügen wir nun noch die Berechtigungen hinzu.

Clients > Client details proxmox-cluster01 OpenID Connect

Clients are applications and services that can request authentication of a user.

Einstellungen Keys Passwörter Rollen Client scopes Authorization Service accounts roles Sessions Advanced

Einstellungen Resources Scopes Policies **Berechtigungen** Evaluate Export





No permissions

If you want to create a permission, please click the button below to create a resource-based or scope-based permission.

[Create resource-based permission](#) [Create scope-based permission](#)

Clients > Client details > Create permission

### Create resource-based permission

Name \* ⓘ proxmox-permission01

Beschreibung ⓘ

Apply to resource type ⓘ Off

Resources \* ⓘ Default Resource X

Policies ⓘ proxmox-zugang01 X

Decision strategy ⓘ  Affirmative  Unanimous  Consensus

**Speichern** **Abbrechen**

## Benutzer Evaluierung

Ein ganz bequemes Werkzeug ist die Benutzer Evaluierung. Dies befindet sich auch in der Clientkonfiguration direkt neben Berechtigungen. Damit ist es möglich Benutzerrechte live zu testen. Da Keycloak einen Cache betreibt, ist das Werkzeug nicht mehr weg zu denken.

Einstellungen Keys Passworter Rollen Client scopes Authorization Service accounts roles Sessions Advanced

Einstellungen Resources Scopes Policies Berechtigungen **Evaluate** Export

Identity Information

Client ⓘ proxmox-cluster01

Benutzer \* ⓘ harald

Rollen ⓘ Select a role

Identity Information

Apply to Resource Type ⓘ Off

Resources and Scopes ⓘ

Schlüssel	Wert
Select or type a key	Select or type a key

Attribut hinzufügen

> Contextual Information

**Evaluate** **Revert**

Hier kann man sehr gut erkennen das der Benutzer „harald“ nicht darf. Sehr gut. Wie sieht es nun aus wenn wir den Benutzer Harald in die Gruppe „gitlab“ werfen?

proxmox-cluster01 (OpenID Connect)

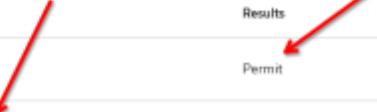
Clients are applications and services that can request authentication of a user.

Einstellungen Keys Passwörter Rollen Client scopes Authorization Service accounts roles Sessions Advanced

Einstellungen Resources Scopes Policies Berechtigungen Evaluate Export

Suche All results ▾

Resource	Overall Results
Default Resource	Permit
Permission	Results
proxmox-permission01	Permit
▪ proxmox-zugang01 voted to PERMIT	Decision strategy
	Unanimous



Und schon darf er sich einloggen.

Wie man sieht ist mit Keycloak/UCS/Proxmox schon einiges möglich. Und damit wäre die Konfiguration auch schon abgeschlossen.

## Automatisch delegierte Berechtigungen über eine LDAP-Gruppe



Damit müssen keine Berechtigungen mehr dem Benutzern in Proxmox manuell zugewiesen werden.



<https://lists.proxmox.com/pipermail/pve-devel/2024-February/061760.html>



Workaround für Gruppenrechte:

<https://docs.software-univention.de/keycloak-app/latest/configuration.html#restrict-access-to-applications>

