



Migration guide: SimpleSAMLPHP to Keycloak

Univention GmbH

Aug 24, 2023

The source of this document is licensed under GNU Affero General Public License v3.0 only.

CONTENTS

1	Limitations of the Keycloak app	3
2	Migration procedure	5
3	Samples for migration of SAML and OIDC services	15
4	Services validation and troubleshooting	23
5	Bibliography	25
	Bibliography	27

This document explains the required steps for the migration from the apps **SimpleSAMLphp** (SAML) and **Kopano Connect** (OIDC) as IdP (Identity Provider) to the app **Keycloak**.

In future releases of UCS (Univention Corporate Server) the **Keycloak** app will replace **SimpleSAMLphp** and the **Kopano Connect** app as the default identity providers in UCS. The reason for this change is that **Keycloak** has many advantages in terms of functionality, configurability and maintainability (e.g. Keycloak provides OIDC and SAML endpoints in one component) over the alternatives.

This documentation is for system administrators who already operate UCS 5.0, make use of the single sign-on features in UCS and want to update their single sign-on configuration to **Keycloak**.

It covers the following topics:

1. *Limitations of the Keycloak app* (page 3)
2. *Migration procedure* (page 5)
3. *Samples for migration of SAML and OIDC services* (page 15)
4. *Services validation and troubleshooting* (page 23)

Warning: The migration from **SimpleSAMLphp** to **Keycloak** is mandatory. **SimpleSAMLphp** is deprecated and planned for removal in a future version of UCS.

This documentation doesn't cover the following topics:

- Detailed information about the usage of the **Keycloak** app, see *Univention Keycloak app documentation* [1]
- Usage of UCS, see *UCS 5.0 Manual* [2].

To understand this documentation, you need to know the following concepts and tasks:

- Use and navigate in a remote shell on Debian GNU/Linux derivative Linux distributions like UCS. For more information, see *Shell and Basic Commands*¹ from *The Debian Administrator's Handbook*, Hertzog and Mas [3].
- *Installation*² and *Configuration*³ of the app **Keycloak** as described in *Univention Keycloak app documentation* [1].
- Know the concepts of SAML (*Security Assertion Markup Language*⁴) and OIDC (*OpenID Connect*⁵) and the differences between the two standards.

Your feedback is welcome and highly appreciated. If you have comments, suggestions, or criticism, please [send your feedback](https://www.univention.com/feedback/?keycloak-migration=generic)⁶ for document improvement.

¹ <https://www.debian.org/doc/manuals/debian-handbook/short-remedial-course.en.html#sect.shell-and-basic-commands>

² <https://docs.software-univention.de/keycloak-app/latest/installation.html#app-installation>

³ <https://docs.software-univention.de/keycloak-app/latest/configuration.html#app-configuration>

⁴ https://en.wikipedia.org/wiki/Security_Assertion_Markup_Language

⁵ [https://en.wikipedia.org/wiki/OpenID#OpenID_Connect_\(OIDC\)](https://en.wikipedia.org/wiki/OpenID#OpenID_Connect_(OIDC))

⁶ <https://www.univention.com/feedback/?keycloak-migration=generic>

LIMITATIONS OF THE KEYCLOAK APP

The **Keycloak** app has not the same feature set as the **SimpleSAMLphp** integration at the time of writing. All of the following points are currently not supported by the **Keycloak** app, but will be implemented in the next versions.

For more information about limitations of the **Keycloak** app, refer to [Requirements and limitations](#)⁷ in *Univention Keycloak app documentation* [1].

1.1 App authorization

In **SimpleSAMLphp** it's possible to restrict the access to certain clients through a checkbox on the user object in the UMC (Univention Management Console). To restrict the access of users to certain clients isn't possible with **Keycloak** at the moment.

1.2 Customization of the login page

With **SimpleSAMLphp** it's possible to adjust the login page through various UCR variables. You can use these UCR variables to add links to the sign-in dialog, for example to redirect the user to the [User self services](#)⁸ in case they forgot their password.

With **Keycloak** you can adjust the theming of the sign-in dialog, but adding custom links isn't supported for the moment.

⁷ <https://docs.software-univention.de/keycloak-app/latest/limitations.html#app-limitations>

⁸ <https://docs.software-univention.de/manual/5.0/en/user-management/user-self-service.html#user-management-password-changes-by-users>

MIGRATION PROCEDURE

Keycloak replaces **SimpleSAMLphp** and **Kopano Konnect** as **SAML IDP**⁹ and **OIDC provider**¹⁰ in a future release of UCS. This section provides a general overview of the migration steps and the required considerations to make before migrating. This migration guide focuses on exclusively on UCS 5.0.

Before the migration can take place, please keep in mind:

- You can migrate services step by step.
- The migration is a manual process.
- Create a backup of the current single sign-on configuration of your services **before** the migration, so that you can rollback in case a problem occurs.
- **SimpleSAMLphp** and **Kopano Konnect** still work even if you installed **Keycloak**.
- After you migrated a service, existing user sessions become invalid. Users have to sign in to the migrated service again.

The migration of one or multiple services always includes at least the following steps:

- The installation and configuration of the **Keycloak** app. For a detailed description, see [Installation](#)¹¹ in the *Univention Keycloak app documentation* [1].
- The creation of **OIDC RP**¹² or **SAML SP**¹³, referred to as *clients* in this document, in **Keycloak** for each service. For more information about how to create those clients, refer to *Migration of services using OIDC for authentication* (page 6) and *Migration of services using SAML for authentication* (page 7).
- The update of the single sign-on configuration of the services to use **Keycloak** as IdP. Have a look at the examples in *Samples for migration of SAML and OIDC services* (page 15).
- The verification that single sign-on works with **Keycloak** as IdP

2.1 Prerequisites

Before the migration can take place, verify to comply to the following prerequisites:

- At least one service uses single sign-on.
- The UCS Primary Directory Node is on UCS version 5.0-4 with the latest errata updates.
- The servers where you want to install **Keycloak** must also be on UCS version 5.0-4 with the latest errata updates.
- If you have **Keycloak** already installed, make sure to update to the latest app version.

⁹ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-IDP>

¹⁰ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-Provider>

¹¹ <https://docs.software-univention.de/keycloak-app/latest/installation.html#app-installation>

¹² <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-RP>

¹³ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

2.2 Migration of services using OIDC for authentication

This section gives a general idea about the migration of services that use **Kopano Konnect** as **OIDC Provider**¹⁴ for the authentication to **Keycloak** as **OIDC Provider**¹⁵.

The general approach for the migration includes the following:

- Install the latest version of the **Keycloak** app in the UCS domain.
- Get an overview of all the services that use **Kopano Konnect** and their settings.
- Create an **OIDC RP**¹⁶, the client, in **Keycloak** for every service that uses **Kopano Konnect** as **OIDC Provider**¹⁷.
- Change the **OIDC** (OpenID Connect) settings in the services to use **Keycloak** as **OIDC Provider**¹⁸ and validate the setup.

To setup a service for **OIDC** with **Keycloak** use the following steps:

1. The **UDM** (Univention Directory Manager) module `oidc/rpservice` configures services that use **Kopano Konnect**. To get a list of all the services and settings, run the following command on the UCS *Primary Directory Node*:

Listing 2.1: List all services that use *Kopano Konnect* for **OIDC**

```
$ udm oidc/rpservice list
```

2. Each service has a `clientid`, a `clientsecret`, and a `redirectURI`. You need the values of these settings to create identical clients for the service in **Keycloak**. On the UCS system where you have installed **Keycloak**, create an **OIDC** client with the following command:

Listing 2.2: Create an *OIDC client* for the service in *Keycloak*

```
$ univention-keycloak oidc/rp create \  
--client-secret clientsecret \  
--app-url redirectURI \  
clientid
```

Note: In case you made custom settings of your **Kopano Konnect** installation, review the following files on your UCS system, that has the app installed:

- `/etc/kopano/konnectd.cfg`
- `/etc/kopano/konnectd-identifier-registration.yaml`
- `/etc/kopano/konnectd-identifier-scopes.yaml`

3. You can also use the **Keycloak Admin Console**¹⁹ to create **OIDC** clients manually or to adjust clients created with `univention-keycloak oidc/rp create`. See also **Keycloak as OpenID Connect provider**²⁰ for more information on how to manage **OIDC** client clients with **Keycloak**.
4. After you created the **OIDC** client for your service, you need to change the **IdP** settings that point to the **OIDC Provider**²¹ in the **OIDC** configuration of the service. Because the services are highly individual in the way they configure **OIDC**, this documentation can't provide a general description. At least, you need the base URL of your **Keycloak** server. Run the following command on the UCS system that has **Keycloak** installed:

¹⁴ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-Provider>

¹⁵ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-Provider>

¹⁶ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-RP>

¹⁷ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-Provider>

¹⁸ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-Provider>

¹⁹ <https://docs.software-univention.de/keycloak-app/latest/installation.html#keycloak-admin-console>

²⁰ <https://docs.software-univention.de/keycloak-app/latest/configuration.html#oidc-op>

²¹ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-Provider>

Listing 2.3: Get base URL of the *Keycloak* server

```
$ univention-keycloak get-keycloak-base-url
```

5. Some services may need to configure the **OIDC Provider**²² URL. Its value is `SSO_URL/realms/ucs`. Replace `SSO_URL` with the output from the previous command.

Other services may have individual settings for the authorization endpoint, the token endpoint, and so on. To get these URLs, run the following commands on the **Keycloak** server.

Listing 2.4: Get different endpoint URLs

```
$ SSO_URL="$(univention-keycloak get-keycloak-base-url)"
$ univention-install jq
$ curl "$SSO_URL/realms/ucs/.well-known/openid-configuration" | jq
{
  "issuer": "https://ucs-sso-ng.example.com/realms/ucs",
  "authorization_endpoint": "https://ucs-sso-ng.example.com/realms/ucs/
  ↪protocol/openid-connect/auth",
  "token_endpoint": "https://ucs-sso-ng.example.com/realms/ucs/protocol/openid-
  ↪connect/token",
  "introspection_endpoint": "https://ucs-sso-ng.example.com/realms/ucs/
  ↪protocol/openid-connect/token/introspect",
  "userinfo_endpoint": "https://ucs-sso-ng.example.com/realms/ucs/protocol/
  ↪openid-connect/userinfo",
  "end_session_endpoint": "https://ucs-sso-ng.example.com/realms/ucs/protocol/
  ↪openid-connect/logout",
  ...
}
```

You don't need to change the settings for the client name and secret, because you have created an OIDC client with identical values in **Keycloak**.

To get a better picture using OIDC with **Keycloak**, have a look at the examples given in section *Services using OIDC* (page 20).

2.3 Migration of services using SAML for authentication

This section gives a general idea about the migration of services that use **SimpleSAMLphp** for the authentication to **Keycloak** as a SAML client.

The general approach for the migration includes the following:

- Install the latest version of the **Keycloak** app in the UCS domain.
- Get an overview of all the services that use **SimpleSAMLphp** and their settings.
- Check and create attribute mappers for LDAP. Selected LDAP attributes become available to **Keycloak**.
- Create a **SAML SP**²³, the client, and necessary attribute mappers in **Keycloak** for every service that uses **SimpleSAMLphp** as **SAML IDP**²⁴.
- Change the SAML (Security Assertion Markup Language) settings in the services to use **Keycloak** as a **SAML IDP**²⁵ and validate the setup.

The following sections explain each step of the migration in detail.

²² <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-Provider>

²³ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

²⁴ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-IDP>

²⁵ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-IDP>

2.3.1 Create LDAP attribute mappers

Some services require specific LDAP attributes, for example, the service requires the user's email address, which the [SAML IDP](#)²⁶ can pass to the service during authentication. Create appropriate LDAP attribute mappers based on your needs.

The user object in **Keycloak** doesn't have every attribute available in the LDAP directory. You can find a list of attributes mapped to **Keycloak** in [Import of user attributes from UCS to Keycloak](#)²⁷ of *Univention Keycloak app documentation* [1].

To check, if the attribute mapping is sufficient for your services, compare the list with the attributes that **SimpleSAMLphp** reads from the LDAP directory. To get a complete list of LDAP attributes available to **SimpleSAMLphp**, run the following command:

Listing 2.5: List IDP LDAP attributes

```
$ udm saml/idpconfig list | sed -n 's/LdapGetAttributes: //p'
```

If you find attributes, that don't map to Keycloak, you can create an LDAP mapper object through the tool **univention-keycloak** or directly in the [Keycloak Admin Console](#)²⁸. To create a mapping on the command line, run the following command:

Listing 2.6: Create LDAP attribute mapping

```
$ univention-keycloak user-attribute-ldap-mapper create $LDAP_ATTRIBUTE_NAME
```

2.3.2 Create a SAML client in Keycloak

Each [SAML SP](#)²⁹, or client, that you have configured for **SimpleSAMLphp**, requires a corresponding [SAML SP](#)³⁰ created in **Keycloak** with the appropriate settings. Depending on the specific settings of your [SAML SP](#)³¹, there are several ways to create a SAML client:

To inspect the **SimpleSAMLphp** configuration on the command line, run:

Listing 2.7: List **SimpleSAMLphp** Provider configuration

```
$ udm saml/serviceprovider list
```

Through the UMC module *SAML Identity Provider*. For a description of each LDAP attribute in this object, see [Adding a new external service provider](#)³² in *UCS 5.0 Manual* [2].

Use one of the following approaches that best suits your needs:

1. If your application provides a metadata XML file for the SAML client settings, you can create a client in **Keycloak** with the command line tool **univention-keycloak** by using either the URL to that XML file or the file itself.

Listing 2.8: Create basic Keycloak client

```
# with URL
$ univention-keycloak saml/sp create --metadata-url="$URL"

# with local XML file
$ univention-keycloak saml/sp create \  

```

(continues on next page)

²⁶ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-IDP>

²⁷ <https://docs.software-univention.de/keycloak-app/latest/configuration.html#ldap-attribute-mapper>

²⁸ <https://docs.software-univention.de/keycloak-app/latest/installation.html#keycloak-admin-console>

²⁹ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

³⁰ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

³¹ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

³² <https://docs.software-univention.de/manual/5.0/en/domain-ldap/saml.html#domain-saml-additional-serviceprovider>

(continued from previous page)

```
--metadata-url="REPLACE WITH SAML CLIENT ISSUER NAME OR URL" \
--metadata-file "/path/to/xml"
```

2. You can transfer each of the SAML client settings from the `saml/serviceprovider` object to the **Keycloak** client. The tool `univention-keycloak` has flags and options for each of these settings.

Table 2.1: Mapping between **SimpleSAMLphp** settings and `univention-keycloak saml/sp create` options

UMC SAML Identity Provider module / UDM attribute name	Keycloak GUI	<code>univention-keycloak saml/sp create</code> option
Service provider activation status: <code>isActive</code>	Enabled	<code>--not-enabled</code>
Service provider identifier: <code>Identifier</code>	Client ID	<code>--client-id</code>
Respond to this service provider URL after login: <code>AssertionConsumerService</code>	Valid redirect URI. This value is also automatically parsed from a metadata XML file.	<code>--valid-redirect-uris</code>
Format of <code>NameID</code> attribute: <code>NameIDFormat</code>	Name ID format	<code>--name-id-format</code>
Description of this service provider: <code>serviceproviderdescription</code>	Description	<code>--description</code>
URL to the service provider's privacy policy: <code>privacypolicyURL</code>	Policy URI	<code>--policy-url</code>
Respond to this service provider URL after login: <code>AssertionConsumerService</code>	Assertion Consumer Service POST Binding URL	<code>--assertion-consumer-url-post</code>
Single logout URL for this service provider: <code>singleLogoutService</code>	Logout Service POST Binding URL	<code>--single-logout-service-url-post</code>
Allow transition of LDAP attributes to the service provider: <code>simplesamlAttributes</code>	n/a	Not available in Keycloak, only implicitly configured if specific attributes are configured to be transitioned to the service provider.
Name of the attribute that is used as <code>NameID</code> : <code>simplesamlNameIDAttribute</code>	n/a	Not available during the creation of the SAML SP ³³ client. Instead another mappings object has to be created for Keycloak , see <i>Configure user attributes for SAML response</i> (page 10).
List of LDAP attributes to transmit: <code>LDAPAttributes</code>	n/a	Not available during the creation of the SAML SP ³⁴ client. Instead another mappings object has to be created for Keycloak , see <i>Configure user attributes for SAML response</i> (page 10).
Value for attribute format field: <code>attributesNameFormat</code>	n/a	Not available during the creation of the SAML SP ³⁵ client. Instead another mappings object has to be created for Keycloak , see <i>Configure user attributes for SAML response</i> (page 10).

³³ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

³⁴ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

³⁵ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

3. You can also use the [Keycloak Admin Console](#)³⁶ to create SAML clients manually or to adjust clients created with `univention-keycloak saml/sp create`. You may want to consult the **Keycloak** documentation at *Managing OpenID Connect and SAML Clients* [4] about how to create clients in Keycloak.
4. Some SAML clients may use custom configurations (e.g. with the attribute `rawsimplesamlSPconfig` of the `saml/serviceprovider` object). If you have a service configured in that way, please check the `/etc/simplesamlphp/metadata.d/*.php` files to get the appropriate settings for the client.

2.3.3 Configure user attributes for SAML response

The following listing demonstrates how to create attribute mappings in **Keycloak** for the `simplesamlNameIDAttribute` and `LDAPattributes` settings of the `saml/serviceprovider` object. These mappings allow you to configure the unique identifier for the user and additional user attributes for the SAML response.

NameID attribute (`simplesamlNameIDAttribute`)

To map the attribute to **Keycloak** that uniquely identifies a user, create a so-called SAML client nameid mapper and attach it to the **SAML SP**³⁷. The table [Table 2.2](#) shows which **SimpleSAMLphp** settings correspond to which options of the command `univention-keycloak saml-client-nameid-mapper create`.

Table 2.2: Mapping between **SimpleSAMLphp** settings and **univention-keycloak saml-client-nameid-mapper create** options

UMC SAML Identity Provider module / UDM attribute name	Keycloak GUI	<code>univention-keycloak saml-client-nameid-mapper create</code> option
Name of the attribute used as <i>NameID</i> : <code>simplesamlNameIDAttribute</code>	User attribute	<code>--user-attribute</code>
Format of <i>NameID</i> attribute: <code>NameID-Format</code>	Mapper <i>NameID</i> format	<code>--mapper-nameid-format</code>

For example, a command might look like one of the following:

³⁶ <https://docs.software-univention.de/keycloak-app/latest/installation.html#keycloak-admin-console>

³⁷ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

Listing 2.9: Create client nameid mapper

```
$ univention-keycloak saml-client-nameid-mapper create CLIENTID MAPPERNAME \
--user-attribute uid \
--mapper-nameid-format urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

Additional user attributes (LDAPattributes)

Some **SAML SP**³⁸ require additional attributes. The name of the attribute on the user object requires a mapping to the name of the attribute in the SAML assertion. The table [Table 2.3](#) shows which **SimpleSAMLphp** settings correspond to which `univention-keycloak saml-client-user-attribute-mapper create` options.

Table 2.3: Mapping between **SimpleSAMLphp** settings and **univention-keycloak saml-client-user-attribute-mapper create** options

UMC SAML Identity Provider module / UDM attribute name	Keycloak GUI	<code>univention-keycloak saml-client-nameid-mapper create</code> option
Value for attribute format field: <code>attributesNameFormat</code>	Name ID	<code>--attribute-nameformat</code>
List of LDAP attributes to transmit: <code>LDAPattributes</code>	User attribute, SAML attribute name	<code>--user-attribute</code> <code>--attribute-name</code>

You need to create such a mapper for each attribute listed in the **SAML SP**³⁹ configuration. In **Keycloak**, you find flags to specify the name of the attribute to send:

--user-attribute

is the attribute name present on the **Keycloak** user object.

--attribute-name

is the SAML attribute utilizing the `urn:oid` namespace.

For example, a command might look like one of the following:

Listing 2.10: Create SAML client user attribute mapper

```
$ univention-keycloak saml-client-user-attribute-mapper create CLIENTID_
↪MAPPERNAME \
--user-attribute uid \
--attribute-name urn:oid:0.9.2342.19200300.100.1.1
```

Note: To get a list of client IDs for all existing **SAML SP**⁴⁰ clients **Keycloak**, run:

³⁸ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

³⁹ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

⁴⁰ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

Listing 2.11: Get a list of client IDs for all existing SAML SP clients

```
$ univention-keycloak saml/sp get
```

2.3.4 Configure SAML services to use Keycloak

After creating [SAML SP](#)⁴¹ clients and mappers in **Keycloak**, the next step is to update the service settings to use **Keycloak** as IdP.

1. Change the SAML settings in the services to use **Keycloak** as [SAML IDP](#)⁴² and validate the setup. The service needs some information about **Keycloak** as the IdP. This includes the *Identity Provider Metadata URL*, and the public certificate.

The SAML Identity Provider Metadata endpoint for Keycloak is `https://$sso_url/realms/ucs/protocol/saml/descriptor`. You can obtain `sso_url` from the output of the following command:

Listing 2.12: Lookup Keycloak base URL

```
$ sso_url="$(univention-keycloak get-keycloak-base-url) "
```

2. It's necessary to update the IdP public certificate in your SAML settings. You can obtain the **Keycloak** server certificate in PEM format with the following command:

Listing 2.13: Retrieve public certificate and *Keycloak* base URL

```
$ univention-keycloak saml/idp/cert get \  
  --as-pem \  
  --output "/tmp/keycloak.cert "
```

How and where you need to configure this information in the service itself is very specific to the service, and this document can't describe it in general terms. Consult the documentation of the service itself. To get a better idea of using SAML with **Keycloak**, take a look at the *Services using SAML* (page 15) in the next section.

2.4 Forward from legacy SimpleSAMLphp to Keycloak

This section is about single sign-on between services that already use **Keycloak** and other services that still use **SimpleSAMLphp**. This configuration can be a temporary solution for environments which have a lot of services to migrate, and where single sign-on between all services needs to be available during the time of the migration. Keep in mind, that the outlined setup is only an short-term solution for your environment until all clients completed the migration to **Keycloak**.

Important: This setup is only important if single sign-on between migrated and not migrated services is needed during the time of the migration. Future releases will not support this setup.

1. Download the certificate from the **Keycloak** app and save it to the local file `/etc/ssl/certs/ucs-sso-ng.keycloak-signing.pem`:

⁴¹ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

⁴² <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-IDP>

Listing 2.14: Download **Keycloak** certificate

```
$ univention-keycloak saml/idp/cert get \
--as-pem \
--output /etc/ssl/certs/ucs-sso-ng.keycloak-signing.pem
```

2. Create a client for **SimpleSAMLphp** in **Keycloak**:Listing 2.15: Create client for **SimpleSAMLphp**

```
$ univention-keycloak saml/sp create \
--umc-uid-mapper \
--metadata-url \
"https://${ucs_sso_fqdn}/simplesamlphp/module.php/saml/sp/metadata.php/default-
→sp" \
--redirect-urls \
"https://${ucs_sso_fqdn}/simplesamlphp/module.php/saml/sp/saml2-acs.php/
→default-sp"
```

3. Change the default provider from `univention-ldap` to `default-sp`:

Listing 2.16: Change default provider

```
$ ucr set saml/idp/authsource=default-sp
```

4. Create a remote IdP for **Keycloak** in **SimpleSAMLphp**:Listing 2.17: Create remote IdP for **Keycloak** in **SimpleSAMLphp**

```
$ kc_provider=$(univention-keycloak get-keycloak-base-url)
$ cat <<EOF > /etc/simplesamlphp/metadata/saml20-idp-remote.php
<?php
\ $metadata['https://${kc_provider}/realms/ucs'] = [
    'SingleSignOnService' => 'https://${kc_provider}/realms/ucs/protocol/saml
→',
    'SingleLogoutService' => 'https://${kc_provider}/realms/ucs/protocol/saml
→',
    'certificate'          => 'ucs-sso-ng.keycloak-signing.pem',
    'authproc' => array(
        50 => array(
            'class' => 'core:AttributeCopy',
            'urn:oid:0.9.2342.19200300.100.1.1' => 'uid',
        ),
    ),
];
EOF
```


SAMPLES FOR MIGRATION OF SAML AND OIDC SERVICES

This section provides detailed examples how to migrate SAML and OIDC services to **Keycloak**.

3.1 Services using SAML

The following examples demonstrate the migration of services that use SAML for authentication and **SimpleSAMLphp** as IdP to **Keycloak** as IdP.

3.1.1 UCS Portal

You can configure the UCS Portal to use SAML for authentication. For a detailed description of how to configure the UCS Portal to use **Keycloak** as a **SAML IDP**⁴³, refer to [Use Keycloak for login to UCS Portal](#)⁴⁴ in the *Univention Keycloak app documentation* [1].

3.1.2 Nextcloud

This section is about migrating the **Nextcloud** app to use **Keycloak** as **SAML IDP**⁴⁵ for the SAML authentication. It assumes that your environment meets the following requirements:

- The configuration of the app **Nextcloud** is complete and done.
- The SAML sign-in for **Nextcloud** works with **SimpleSAMLphp** as **SAML IDP**⁴⁶.
- The UCS domain has the latest version of the app **Keycloak** installed.

To setup **Nextcloud** for SAML with **Keycloak**, use the following steps:

1. To create a SAML client for **Nextcloud**, run the following commands on the UCS system that has the **Keycloak** app installed. Replace `NEXTCLOUD_FQDN` with the FQDN (fully qualified domain name) of the UCS system that has **Nextcloud** installed.

Listing 3.1: Create SAML client for *Nextcloud* in *Keycloak*

```
$ NEXTCLOUD_FQDN="REPLACE with NEXTCLOUD_FQDN"
$ univention-keycloak saml/sp create \
  --metadata-url="https://$NEXTCLOUD_FQDN/nextcloud/apps/user_saml/saml/
↪metadata" \
  --role-mapping-single-value
```

2. To obtain the **Keycloak** base URL and certificate, run the following commands on the UCS system that has **Keycloak** installed:

⁴³ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-IDP>

⁴⁴ <https://docs.software-univention.de/keycloak-app/latest/configuration.html#login-portal>

⁴⁵ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-IDP>

⁴⁶ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-IDP>

Listing 3.2: Retrieve public certificate of *Keycloak*

```
$ univention-keycloak saml/idp/cert get \  
  --as-pem \  
  --output "/tmp/keycloak.cert"
```

The output of the command in [Listing 3.2](#) saves the certificate in the file `/tmp/keycloak.cert`. Copy this file to the UCS system that has the **Nextcloud** app installed.

Listing 3.3: Retrieve public certificate and *Keycloak* base URL

```
$ univention-keycloak get-keycloak-base-url
```

The command in [Listing 3.3](#) outputs the base URL of your **Keycloak** server. Replace `SSO_URL` in the following instruction with this value.

3. To change the IdP settings for **Nextcloud**, run the following commands on the UCS system that has it installed. Copy the certificate file `/tmp/keycloak.cert` from [Listing 3.2](#) to the UCS system with **Nextcloud** installed and replace `SSO_URL`.

Listing 3.4: Configure the **Nextcloud** app to use **Keycloak** as IdP

```
$ SSO_URL="REPLACE WITH SSO_URL"  
$ univention-app shell nextcloud sudo -u www-data /var/www/html/occ_\  
↪saml:config:set \  
  --idp-x509cert="$(cat /tmp/keycloak.cert)" \  
  --general-uid_mapping="uid" \  
  --idp-singleLogoutService.url="$SSO_URL/realms/ucs/protocol/saml" \  
  --idp-singleSignOnService.url="$SSO_URL/realms/ucs/protocol/saml" \  
  --idp-entityId="$SSO_URL/realms/ucs" 1
```

To validate the setup, visit the sign-in page of your **Nextcloud** app and initiate a single sign-on. **Nextcloud** redirects you to **Keycloak** for authentication. You can use **Nextcloud** after authentication.

See also:

[Nextcloud](#)⁴⁷

in Univention App Catalog

3.1.3 Google Workspace Connector

This section provides a step-by-step guide for migrating the **Google Connector** app to use **Keycloak** as **SAML IDP**⁴⁸. The migration assumes that your environment meets the following requirements:

- The configuration of the app **Google Workspace Connector** is complete and done.
- The SAML login for your *Google Workspace* works with **SimpleSAMLphp**.
- You have the administrator credentials for your *Google Workspace Admin Console* for the SAML service configuration.
- The UCS domain has the latest version of the app **Keycloak** installed.

To setup **Google Workspace Connector** for SAML with **Keycloak** use the following steps:

1. Create a **SAML SP**⁴⁹, the client, for **Google Workspace Connector** in **Keycloak**. Run the following commands on the UCS system that has **Keycloak** installed:

⁴⁷ <https://www.univention.com/products/univention-app-center/app-catalog/nextcloud/>

⁴⁸ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-IDP>

⁴⁹ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

Listing 3.5: Create SAML client for *Google Workspace Connector* in *Keycloak*

```

$ google_domain="REPLACE_WITH_NAME_OF_YOUR_GOOGLE_DOMAIN"
$ univention-keycloak saml/sp create \
  --client-id google.com \
  --assertion-consumer-url-post "https://www.google.com/a/$google_domain/acs
→" \
  --single-logout-service-url-post "https://www.google.com/a/$google_domain/
→acs" \
  --idp-initiated-sso-url-name google.com \
  --name-id-format email \
  --frontchannel-logout-off
$ univention-keycloak user-attribute-ldap-mapper \
  create univentionGoogleAppsPrimaryEmail
$ univention-keycloak saml-client-nameid-mapper create \
  google.com \
  univentionGoogleAppsPrimaryEmail \
  --user-attribute univentionGoogleAppsPrimaryEmail \
  --mapper-nameid-format urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress

```

2. For the configuration of the SAML settings of your *Google Workspace* you need the public certificate and the base URL of the **Keycloak** server. Run the following commands on the UCS system that has **Keycloak** installed:

Listing 3.6: Retrieve public certificate and *Keycloak* base URL

```

$ univention-keycloak saml/idp/cert get --as-pem --output /tmp/keycloak.cert

```

The output of the command in [Listing 3.6](#) is the certificate. Copy the file `/tmp/keycloak.cert` to your local client computer.

Listing 3.7: Retrieve public certificate and *Keycloak* base URL

```

$ univention-keycloak get-keycloak-base-url

```

The command in [Listing 3.7](#) outputs the base URL of your **Keycloak** server. Replace `SSO_URL` in the following instruction with this value.

3. Change the *Third-party SSO profile for your organisation* settings in the *Google Workspace Admin console* of your google domain.
 1. Open the URL <https://admin.google.com> and login with your administrator account.
 2. Go to *Security* ▶ *Authentication* ▶ *SSO with third-party IdP*.
 3. Open *Third-party SSO profile for your organisation*.
 4. Change *Sign-in page URL* to `SSO_URL/realms/ucs/protocol/saml`.
 5. Change *Sign-out page URL* to `SSO_URL/realms/ucs/protocol/openid-connect/logout`.
 6. To upload the **Keycloak** certificate in `/tmp/kc.cert` from [Listing 3.6](#), click *REPLACE CERTIFICATE*.
 7. To activate the settings, click *Save*.
4. Change the link in the UCS portal entry *Google Workspace login* for the IdP initiated single sign-on. On your UCS *Primary Directory Node* run the following commands:

Listing 3.8: Change portal entry for *Google Workspace login* to IdP initiated single sign-on

```
$ google_domain="REPLACE WITH NAME_OF_YOUR_GOOGLE_DOMAIN"
$ SSO_URL="REPLACE WITH SSO_URL"
$ udm portals/entry modify \
--dn "cn=SP,cn=entry,cn=portals,cn=univention,$(ucr get ldap/base)" \
--set link="'en_US' '$SSO_URL'/realms/ucs/protocol/saml/clients/google.
↪com?RelayState=https://www.google.com/a/'$google_domain'/ServiceLogin''
```

To validate the setup, visit <https://google.com> and sign in with one of the UCS user accounts enabled for *Google Workspace*. Also, verify the UCS portal entry *Google Workspace login* for the IdP initiated single sign-on.

Warning: The automatic redirect after the single sign-out doesn't work with **Keycloak**.

See also:

Google Workspace Connector^{Page 18, 50}
in Univention App Catalog

3.1.4 Microsoft 365 Connector

The example illustrates how to migrate the app **Microsoft 365 Connector** to use **Keycloak** as **SAML IDP**⁵¹. It assumes that your environment meets the following requirements:

- The configuration of the app **Microsoft 365 Connector** is complete and done.
- The SAML login for your *Azure Active Directory* domain works with **SimpleSAMLphp**.
- You have a client computer with *Microsoft Windows* installed on it and a working internet connection on the client computer to configure SAML in *Azure Active Directory*.
- You have the *Administrator* credentials for your *Azure Active Directory* domain for the SAML service configuration.
- The UCS domain has the latest version of the app **Keycloak** installed.

To setup **Microsoft 365 Connector** for SAML with **Keycloak**, use the following steps:

1. Create a **SAML SP**⁵², the client, for **Microsoft 365 Connector** in **Keycloak**. Run the following commands on the UCS system that has **Keycloak** installed:

Listing 3.9: Create SAML client for *Microsoft 365 Connector* in *Keycloak*

```
# get the saml client metadata xml from microsoft
$ curl https://nexus.microsoftonline-p.com/federationmetadata/saml20/
↪federationmetadata.xml > /tmp/ms.xml

# create the client in keycloak
$ univention-keycloak saml/sp create \
--metadata-file /tmp/ms.xml \
--metadata-url urn:federation:MicrosoftOnline \
--idp-initiated-sso-url-name MicrosoftOnline \
--name-id-format persistent

# create a SAML nameid mapper
$ univention-keycloak saml-client-nameid-mapper create \
```

(continues on next page)

⁵⁰ <https://www.univention.com/products/univention-app-center/app-catalog/google-apps/>

⁵¹ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-IDP>

⁵² <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

(continued from previous page)

```
urn:federation:MicrosoftOnline \
entryUUID \
--mapper-nameid-format "urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
↪" \
--user-attribute entryUUID \
--base64
```

2. For the configuration of the SAML settings of your *Azure Active Directory* domain you need the public certificate and the base URL of the **Keycloak** server. Run the following commands on the UCS system that has **Keycloak** installed:

Listing 3.10: Retrieve public certificate and *Keycloak* base URL

```
$ univention-keycloak saml/idp/cert get --output /tmp/keycloak.cert
$ cat /tmp/keycloak.cert
```

The output of the command in Listing 3.10 is the certificate. Replace *KEYCLOAK_CERTIFICATE* in the following steps with this value.

Listing 3.11: Retrieve public certificate and *Keycloak* base URL

```
$ univention-keycloak get-keycloak-base-url
```

The command in Listing 3.11 outputs the base URL of your **Keycloak** server. Replace *SSO_URL* in the following steps with this value.

3. To configure the SAML settings for the *Azure Active Directory* domain, copy the following code block to your *Microsoft Windows* client computer. Replace the values for *SSO_URL*, *KEYCLOAK_CERTIFICATE*, the *Azure Active Directory* domain name and credentials and run the script in the *Microsoft Windows PowerShell*.

Listing 3.12: Change *Azure Active Directory* domain authentication to use *Keycloak*

```
# CHANGE this according to your setup/environemt
$sso_url = "replace with SSO_URL"
$signing_cert = "replace with KEYCLOAK_CERTIFICATE"
$domain = "YOUR AZURE DOMAIN NAME"
$username = "YOUR AZURE DOMAIN ADMIN"
$password = "PASSWORD OF YOUR AZURE DOMAIN ADMIN"
# CHANGE end

$issuer_uri = "$sso_url/realms/ucs"
$logon_uri = "$sso_url/realms/ucs/protocol/saml"
$passive_logon_uri = "$sso_url/realms/ucs/protocol/saml"
$logoff_uri = "$sso_url/realms/ucs/protocol/saml"
$pass = ConvertTo-SecureString -String "$password" -AsPlainText -Force
$credential = New-Object -TypeName System.Management.Automation.PSCredential -
↪ArgumentList $username, $pass
$o365cred = Get-Credential $credential

Install-Module MSOnline
Import-Module MSOnline
Connect-MsolService -Credential $o365cred

Set-MsolDomainAuthentication -DomainName "$domain" -Authentication Managed
Set-MsolDomainAuthentication `
  -DomainName "$domain" `
  -FederationBrandName "UCS" `
  -Authentication Federated `
  -ActiveLogOnUri "$logon_uri" `
```

(continues on next page)

(continued from previous page)

```
-PassiveLogOnUri "$passive_logon_uri" \  
-SigningCertificate "$signing_cert" \  
-IssuerUri "$issuer_uri" \  
-LogOffUri "$logoff_uri" \  
-PreferredAuthenticationProtocol SAML  
  
Get-MsolDomain  
Pause
```

4. To change the link in the UCS portal entry *Microsoft 365 Login* for the IdP initiated single sign-on, run the following commands on your UCS *Primary Directory Node*:

Listing 3.13: Change portal entry for *Microsoft 365 Login* to IdP initiated single sign-on

```
$ SSO_URL="REPLACE WITH SSO_URL"  
$ udm portals/entry modify \  
  --dn "cn=office365,cn=entry,cn=portals,cn=univention,$(ucr get ldap/base)" \  
  --set link="'en_US' '$SSO_URL'/realms/ucs/protocol/saml/clients/  
  ↪MicrosoftOnline'
```

To validate the setup, visit <https://www.microsoft365.com/> and sign in with one of the UCS user accounts enabled for *Microsoft 365*. Also, verify the UCS portal entry *Microsoft 365 Login* for the IdP initiated single sign-on.

Warning: The automatic redirect after the single sign-out doesn't work with **Keycloak**.

See also:

[Microsoft 365 Connector](#)^{Page 20, 53}
in Univention App Catalog

3.2 Services using OIDC

The following examples demonstrate the migration of services that use OIDC for authentication and **Kopano Connect** as IdP to **Keycloak** as IdP.

3.2.1 ownCloud

This section is about the migration of the **ownCloud** app to use **Keycloak** as **OIDC Provider**⁵⁴ for authentication. It assumes that your environment meets the following requirements:

- The configuration of the app **ownCloud** is complete and done.
- The OIDC sign-in for **ownCloud** works with **Kopano Connect** as **OIDC Provider**⁵⁵.
- The UCS domain has the latest version of the app **Keycloak** installed.

To setup **ownCloud** for OIDC with **Keycloak** use the following steps:

1. To obtain the necessary information such as `clientsecret` and `redirectURI`, run the following command on the UCS *Primary Directory Node*. You need the values to create the **OIDC RP**⁵⁶, the client, in the next step.

⁵³ <https://www.univention.com/products/univention-app-center/app-catalog/microsoft365/>

⁵⁴ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-Provider>

⁵⁵ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-Provider>

⁵⁶ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-RP>

Listing 3.14: Get current settings for the *ownCloud* OIDC client

```
$ udm oidc/rpservice list --filter name=owncloud
DN: cn=owncloud,cn=oidc,cn=univention,dc=...
  applicationtype: web
  clientid: owncloud
  clientsecret: -> copy this value
  insecure: None
  name: owncloud
  redirectURI: -> copy this value
  trusted: yes
```

Look for the values of `clientsecret` and `redirectURI` and copy them, for example, into a temporary text file.

2. To create the **OIDC RP**⁵⁷, the client, for **ownCloud** in **Keycloak**, run the following commands on the UCS system that has **Keycloak** installed. Replace `clientsecret` and `redirectURI` with the values for these settings in Listing 3.14 from the previous step.

Listing 3.15: Create OIDC client for *ownCloud* in *Keycloak*

```
$ CLIENT_SECRET="REPLACE WITH clientsecret"
$ REDIRECT_URI="REPLACE WITH redirectURI"
$ univention-keycloak oidc/rp create \
  --client-secret "$CLIENT_SECRET" \
  --app-url "$REDIRECT_URI" owncloud
```

3. To obtain the base URL of your **Keycloak** server, run the following command on the UCS system that has it installed:

Listing 3.16: Obtain *Keycloak* base URL

```
$ univention-keycloak get-keycloak-base-url
```

Replace `SSO_URL` in the following step with this value.

4. Change the IdP setting in **ownCloud**. Run the following command on the UCS system that has **ownCloud** installed:

Listing 3.17: Change IDP settings in *ownCloud*

```
$ SSO_URL="REPLACE WITH SSO_URL"
$ univention-app configure owncloud \
  --set OWN_CLOUD_OPENID_PROVIDER_URL="$SSO_URL/realms/ucs"
```

To validate the setup, visit the sign-in page of your **ownCloud** app and initiate a single sign-on. **ownCloud** redirects you to **Keycloak** for authentication. You can use **ownCloud** after authentication.

See also:

ownCloud^{Page 21, 58}

in Univention App Catalog

⁵⁷ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-OIDC-RP>

⁵⁸ <https://www.univention.com/products/univention-app-center/app-catalog/owncloud/>

SERVICES VALIDATION AND TROUBLESHOOTING

Every procedure during the migration has the potential to fail at some point. This section provides hints for troubleshooting such situations.

4.1 Log files

As first step, review the log files of **Keycloak** and the connected services:

- Review the log files of **Keycloak** and look for errors. On the UCS system that has **Keycloak** installed, run the following command:

```
$ univention-app logs keycloak
```

- For an extensive troubleshooting guide of the **Keycloak** app, refer to [Troubleshooting⁵⁹](#) in the *Univention Keycloak app documentation* [1].
- Review the log files of the following services:
 - For UMC: `/var/log/univention/management-console-web-server.log`
 - For **Nextcloud**: `/var/lib/univention-appcenter/apps/nextcloud/data/nextcloud-data/nextcloud.log`
 - For **ownCloud**: `/var/lib/univention-appcenter/apps/owncloud/data/files/owncloud.log`
 - For other services: consult the manual of the service

4.2 Single sign-on settings

Also verify the configuration of **Keycloak** and the single sign-on settings of the services:

- For OIDC services make sure that the service has the correct `clientsecret` and `clientid`. The values for these settings must match in **Keycloak** and the services.
- For SAML services verify that the service uses the current, public certificate of the **Keycloak** server.
- For SAML verify that the `clientid` in the **Keycloak** configuration of your **SAML SP⁶⁰** is correct. This is also the issuer for the SAML authentication request. The value is service specific, but needs to match the expectations of the service.

Additionally, verify the following items:

- Ensure that all involved systems have the same and synchronized time.
- Use *Developer Tools* of your browser to see which requests fail to narrow down the cause of the problem.

⁵⁹ <https://docs.software-univention.de/keycloak-app/latest/troubleshooting.html#app-troubleshooting>

⁶⁰ <https://docs.software-univention.de/keycloak-app/latest/architecture.html#term-SAML-SP>

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] *Univention Keycloak app documentation*. Univention GmbH, 2023. URL: <https://docs.software-univention.de/keycloak-app/latest/>.
- [2] *UCS 5.0 Manual*. Univention GmbH, 2021. URL: <https://docs.software-univention.de/manual/5.0/en/>.
- [3] Raphaël Hertzog and Roland Mas. *The Debian Administrator's Handbook*, chapter Shell and Basic Commands. Freexian SARL, First edition, 2020. URL: <https://www.debian.org/doc/manuals/debian-handbook/short-remedial-course.en.html#sect.shell-and-basic-commands>.
- [4] *Managing OpenID Connect and SAML Clients*. URL: https://www.keycloak.org/docs/21.1.2/server_admin/#assembly-managing-clients_server_administration_guide.