

**HCL MT CH-MSP Product Documentation**

**Keycloak Authentication and SSO**

**SSO and Authenticate using Customer IdP**

* **Config keycloak to use Customer IdP using SAML**

**Configure KeyCloak as OIDC provider for Connections**

**SSO between Connections, SameTime and Domino**

**Configure KeyCloak to connect to an LDAP**

**Configure WAS as OIDC RP**

* **Configure WAS as OIDC RP in multi-clusters env**
* **multi-clusters- make sure set- Trusted authentication realms - outbound**

**Configure Connections Application user roles**

**update reverse proxy to handle some redirects**

**Support OAuth 2 for internal apps and 3rd party apps**

* **Allowing third-party applications access to data via the OAuth2 protocol**
* **Enable EE using OIDC OP as OAuth2 provider**
* **Enable RTE using OIDC OP as OAuth Provider**

**Keycloak Authentication and SSO**

Note: the dot line from connections to ldap is not required for authentication. It may be used for group support from application level. Although it should really be done using keycloak APIs, the existing code may be doing a direct ldap call.



In MT, most of customers have their own IdP to authenticate users so Connections can SSO to customers own applications that use the same IdP.

We can use Keycloak to connect to customer Idp either via SAML or OIDC.



By default, user would have to pick which login to use. However we can remove this page by redirect the user to their org's IdP.

This can be done by using **kc\_idp\_hint**. see <https://www.keycloak.org/docs/latest/server_admin/#_client_suggested_idp>

Because redirect URL contains org url, so we can create rewrite rule to map between org to their own IdP.

https://\

{KK}/realms/{RM}/protocol/openid-connect/auth?xxxxx&redirect\_uri=https%3A%2F%2Fmtdemo1- orgc.cnx.cwp.pnp-hcl.com%3A443%2Foidcclient\_apps%2Fkeycloak&yyyyy

to https://\{KK}

/realms/{RM}/protocol/openid-connect/auth?xxxxx&redirect\_uri=https%3A%2F%2Fmtdemo1-orgc.cnx.cwp.pnp-

hcl.com%3A443%2Foidcclient\_apps%2Fkeycloak&yyyyy&kc\_idp\_hint=mtdemo1-orgc

Here is an example implementation:

if ($arg\_redirect\_uri ~ ^(https.\*connmt-orge.\*)){

rewrite ^(/auth/.\*)/Azure-OIDC/(\w+\.?.\*$) $1/Azure-OIDC/$2?kc\_idp\_hint=google break;

}

if ($arg\_redirect\_uri ~ ^(https.\*connmt-orgf.\*)){

rewrite ^(/auth/.\*)/Azure-OIDC/(\w+\.?.\*$) $1/Azure-OIDC/$2?kc\_idp\_hint=connmt-orgf break;

}

**Config keycloak to use Customer IdP using SAML**

Azure AD as SAML IdP

**Configure KeyCloak as OIDC provider for Connections**

**Create client for Connections web client**

Create a realm

Connect to the LDAP that contains all Connections users Create an OIDC client, type: **confidential**

In Mapper add properties name: realmName, hardcode the value as your realm name

**Note, you can add the mapper in the client scope so it will be available to all clients in the same realm including web, mobile, desktop, conn-ee, conn-rte, and 3rd party clients.**



call back urls:

[https://<connections host>/oidcclient/<](https://connmt-orga.cnx.cwp.pnp-hcl.com/oidcclient/keycloak)provider\_1.identifier value> Note: for MT also add each org's url, e.g.

[https://<connections host>\_orga/oidcclient/<](https://connmt-orga.cnx.cwp.pnp-hcl.com/oidcclient/keycloak)provider\_1.identifier value> [https://<connections host>\_orgb/oidcclient/<](https://connmt-orga.cnx.cwp.pnp-hcl.com/oidcclient/keycloak)provider\_1.identifier value>

**Create client for Connections Mobile client**

In the same realm, create another client for Connections Mobile:



**SSO between Connections, SameTime and Domino can be done via Keycloak.**

All apps should be in the same realm. Create each client for each app.

**The client for SameTime: use IdP initiated Post binding with signed assertion:**





**The client for Connections is OIDC:**

https://apps.na.collabserv.com/wikis/home?lang=en- us#!/wiki/W0d07dd0b225e\_410e\_a5c4\_1b9cfc43101d/page/Configure%20KeyCloak%20as%20OIDC%20provider%20for%20Connections

