IPA-TUURA
FreeIPA connector for Keycloak
4 February 2024 | Brussels, Belgium

Francisco Triviño <ftrivino@redhat.com>
Identity and access management is an umbrella term, currently it defines multiple technologies and business processes to access the right assets at the right time for the right reasons while keeping an authorized access. Some examples of IAM products are:

- Microsoft Active Directory
- Red Hat Identity Management (FreeIPA)
- Keycloak
- Okta
- EntraID
- …..
Background

FreeIPA and Keycloak

FreeIPA

- Integrated identity management solution for POSIX-like environments (linux)
- Users and Groups consumed by the applications running in POSIX environment
- Ability to run application processes in presence of POSIX user and group IDs.

Keycloak

- IAM for Modern Applications
- Application level identities are not necessarily the same the system level ones.

Active Directory

- Users and groups relies on Security Identifiers (SIDs)
- Organizational Units (OUs)
Sometimes you need to integrate multiple IAM solutions

Sometimes you are happy having a standalone IAM solution... but that’s not the usual case... IAM defines:

- **SSO**: access multiple applications within the same organization or domain using a single set of credentials
- **Identity and User Federation**: It enables users to access applications or platforms across multiple enterprise domains that are part of the federated configuration

Source:
Keycloak User Federation Storage

- Keycloak first checks its internal user store when a user logs in and then looks through configured external User Storage providers if needed. Data from external stores is mapped into a common user model for runtime use.
- Keycloak already supports integration with FreeIPA as a backend to lookup and authenticate identities.

Source: https://www.keycloak.org/docs/latest/server_admin/#_user-storage-federation
Ipa-tuura: FreeIPA connector for Keycloak

Background

Have a look to existing integration

Keycloak and LDAP/AD

▸ Keycloak includes an LDAP/AD provider. You can federate multiple different LDAP servers in one Keycloak realm and map LDAP user attributes into the Keycloak common user model.

Keycloak and SSSD/IdM

▸ Keycloak includes the System Security Services Daemon (SSSD) plugin. SSSD is part of the Fedora and Red Hat Enterprise Linux (RHEL), and it provides access to multiple identities and authentication providers. SSSD also provides benefits such as failover and offline support.

Source:
https://www.keycloak.org/docs/latest/server_admin/#_ldap
https://www.keycloak.org/docs/latest/server_admin/#_sssd
What's the problem then?
What problems are we trying to solve

Keycloak and LDAP/SSSD/FreeIPA

Existing gaps of current Keycloak integration with IdM/SSSD/LDAP

- SSSD/IdM and LDAP/AD integrations offer different features - missing feature parity
- Existing SSSD federation plugin is read-only, requires java dbus libraries and UNIX sockets
- Limitations for deployment in containers
- Complicated setup steps required
What problems are we trying to solve
Keycloak and LDAP/SSSD/FreeIPA

Existing gaps of current Keycloak integration with IdM/SSSD/LDAP

- SSSD/IdM and LDAP/AD integrations offer different features - missing feature parity
- Existing SSSD federation plugin is read-only, requires java dbus libraries and UNIX sockets
- Limitations for deployment in containers
- Complicated setup steps required

Re-design is needed
New ipa-tuura service comes into the play
What about to make a generic bridge?

Integration, automation, security, scalability....

We need a common API for managing identities, among other requirements:

▸ Able to read and write, authenticate users, from an Integration Domain

▸ Simplify integration. Replace existing plugins by just 1 plugin for FreeIPA/AD/LDAP

▸ Easy management of users/groups using the available lookup and import strategies

▸ Cloud-friendly maintainable solution

▸ No performance impact

▸ Do not reinvent the wheel, rely on existing open source projects
...... start from scratch?

- No need to...
  - There are multiple existing SCIM v2 open source projects we can rely on
- Let’s choose django-scim2,
  - written in Python, similar to FreeIPA.

SCIM 2.0 is released as RFC7642, RFC7643 and RFC7644 under IETF

- RFC7643 - SCIM: Core Schema
- RFC7644 - SCIM: Protocol
- RFC7642 - SCIM: Definitions

REST API CRUD operations

- POST
- PUT
- GET
- DELETE
Ipa-tuura architecture

...... Cloud-friendly maintainable solution

FROM quay.io/centos/centos:stream9
...
...

freeipa/
ipa-tuura

Keycloak

django

SCIM

FreeIPA
Ipa-tuura architecture

...... Security

FROM quay.io/centos/centos:stream9

APACHE

Keycloak <-> HTTPs <-> django <-> SCIM <-> FreelIPA
Ipa-tuura architecture

...... generic API
Ipa-tuura architecture

...... generic API
Ipa-tuura architecture

... no performance impact

FROM quay.io/centos/centos:stream9

HTTPs

APACHE

REQUEST

RESPONSE

django

REST framework

FreeIPA API

WRITE

READ

FreeIPA

cache

sssd

D-Bus infopipe

Ipa-tuura: FreeIPA connector for Keycloak
Ipa-tuura architecture

... unify

Ipa-tuura: FreeIPA connector for Keycloak

FROM quay.io/centos/centos:stream9

APACHE

Keycloak

HTTPs

REQUEST

RESPONSE

FreeIPA

Python LDAP

Active Directory

READ

WRITE

FreeIPA API

D-Bus infopipe

cache

s3sd

Keycloak

REST

django

framework

django

SCIM

Ipa-tuura: FreeIPA connector for Keycloak
What about Keycloak, does it support SCIM calls?
What about Keycloak?

Replace existing SSSD plugin by a generic SCIM Client

Replace existing plugin with SCIM plugin for IdM/AD/LDAP:

- New Keycloak plugin acts as SCIM client, uses Apache HTTPs client to make calls to scim v2 endpoints
- Requests for user information and user authentication in keycloak will be forwarded to the plugin and proxied to backend
  - Users SSO login and password authentication
  - Supports
  - -- Looking up users
  - -- Adding users
  - -- Deleting users
  - -- User modifications (Email, first name, last name)

Source: [https://github.com/justin-stephenson/scim-keycloak-user-storage-spi/](https://github.com/justin-stephenson/scim-keycloak-user-storage-spi/)
What about Keycloak?

Replace existing SSSD plugin by a generic SCIM Client

SCIMv2 Bridge connection

Integration Domain enrollment

SCIMv2 Bridge options
(configures SSSD)
Summing up

New integration

- Keycloak plugin sends with ipa-tuura (Bridge service) over HTTPS to /domains/v1/domain endpoint to add and remove Integration domains.
- Keycloak plugin communicates with ipa-tuura (Bridge service) over HTTPS to /scim/v2 specification endpoints.
- ipa-tuura provides REST API, translates SCIM endpoint requests into identity provider operations on the backend.
- Keycloak Plugin does not communicate directly with backend servers (IDM, AD, LDAP).
Add FreeIPA Integration Domain

1) HTTPs POST request to `/domains/v1/Domain`
   - `bridge.ipa.test:4430`
   - `ou=people,dc=ipa,dc=test`
   - ...

2) configure SSSD with IPA provider
3) add ipa-tuura `service` using IPAs API
4) add ipa-tuura `role` using IPAs API
5) add ipa-tuura `privilege` using IPAs API
6) generate a `keytab` for the writable interface
Ipa-tuura: FreeIPA connector for Keycloak

Ready to manipulate users

**DEMO**

User Federation

keycloak.ipa.test

bridge.ipa.test

idm.ipa.test

▸ /scim/v2/Users POST

```
{
  "userName": "testuser",
  "emails": [
    {
      "primary": true,
      "type": "work",
      "value": "testuser@ipa.test"
    }
  ],
  "name": {
    "formatted": "testuser",
    "familyName": "user",
    "givenName": "test"
  },
  "externalId": "testuser",
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "meta": {
    "resourceType": "User"
  }
}
```

▸ IPAs API

```
api.command.user_add("ipauser", givenname="ipa", sn="user")
"result": {
  "displayName": ["ipa user"],
  "cn": ["ipa user"],
  "uuid": ["ippass@ipa.test"],
  "krbprincipalname": ["ipapython.kerberos.Principal("test@IPA.TEST")"],
  "loginshell": ["/bin/sh"],
  "initials": ["lu"],
  "uid": ["ipauser"],
  "uidnumber": ["1445000004"],
  ...
  ...
  "dn": "ipapython.dn.ON(\"uid=test,cn=users,cn=accounts,dc=ipa,dc=test\")",
}```
Ipa-tuura: FreeIPA connector for Keycloak
Work In Progress
Kerberos GSSAPI Authentication

keycloak.ipa.test

bridge.ipa.test

idm.ipa.test

1) /bridge/login_kerberos/

2) mod_auth_gssapi, ipa keytab

3) Response session_cookie

4) Response session_cookie

User Federation

SCIM

FreeIPA

SCIM
Potential usages

The bridge can also be used in a variety of different scenarios:

- Synchronization of identities across different providers
- Migration of identities across different providers
- Provide a SCIM server for other IAMs such as Okta, EntraID...

https://github.com/freeipa/ipa-tuura

https://github.com/justin-stephenson/scim-keycloak-user-storage-spi/
Tech Stack

Q&A

Source: Tech Stack

ipa-tuura: FreeIPA connector for Keycloak

version number here V00000
Thank you