Keycloak

Keycloak est une solution open source de gestion des identités et des accès. Elle permet de centraliser l’authentification des utilisateurs à travers plusieurs applications, en proposant notamment une connexion unique appelée **SSO** (Single Sign-On). Grâce à Keycloak, les utilisateurs peuvent se connecter une seule fois pour accéder à plusieurs services, sans avoir à ressaisir leurs identifiants à chaque fois.

Keycloak est conçu pour être intégré facilement à des applications web, des API, ou des services internes, tout en garantissant un haut niveau de sécurité. Il peut s’appuyer sur des annuaires existants comme LDAP ou Active Directory, et permet aussi la connexion via des comptes sociaux (Google, GitHub, etc.).

Présentation

Qu’est-ce que Keycloak ?

Keycloak est une solution open source de gestion des identités et des accès. Elle permet de centraliser l’authentification des utilisateurs à travers plusieurs applications, en proposant notamment une connexion unique appelée **SSO** (Single Sign-On). Grâce à Keycloak, les utilisateurs peuvent se connecter une seule fois pour accéder à plusieurs services, sans avoir à ressaisir leurs identifiants à chaque fois.

Keycloak est conçu pour être intégré facilement à des applications web, des API, ou des services internes, tout en garantissant un haut niveau de sécurité. Il peut s’appuyer sur des annuaires existants comme LDAP ou Active Directory, et permet aussi la connexion via des comptes sociaux (Google, GitHub, etc.).

À quoi sert Keycloak ?

Keycloak sert principalement à gérer les connexions, les utilisateurs, et leurs droits d’accès dans un environnement informatique. Il simplifie la vie des utilisateurs en leur évitant de retenir plusieurs mots de passe, et il facilite le travail des administrateurs en centralisant la gestion des comptes et des permissions.

C’est une brique essentielle dans toute architecture qui doit gérer plusieurs services, tout en assurant la sécurité et la traçabilité des connexions. Keycloak est également utilisé pour appliquer une authentification forte (2FA), ou pour créer des portails d'accès unifiés à des ressources internes.

Comment fonctionne Keycloak ?

Keycloak fonctionne comme un serveur central d’identités. Les applications clientes ne gèrent pas directement les utilisateurs : elles délèguent cette tâche à Keycloak, via des standards comme

OAuth2, OpenID Connect ou SAML.

Lorsqu'un utilisateur tente d'accéder à une application, il est redirigé vers Keycloak, qui vérifie son identité. Une fois connecté, Keycloak transmet à l'application les informations nécessaires (nom, e-mail, rôles, etc.), sans que le mot de passe transite directement.

Il peut aussi se connecter à une base LDAP existante, ou permettre la création de comptes internes. Les rôles et groupes permettent de limiter les droits d'accès selon les besoins.

Pourquoi utiliser Keycloak ?

Keycloak est une solution idéale pour ceux qui cherchent à gérer plusieurs applications avec une authentification centralisée. Il évite la duplication des comptes, renforce la sécurité grâce à la double authentification, et permet une meilleure traçabilité.

L'interface d'administration est simple à prendre en main, et l'outil s'intègre avec de nombreuses technologies existantes. De plus, étant open source, il peut être **auto-hébergé**, ce qui garantit un contrôle total sur les données d'authentification.

Cas d'usage courants

Keycloak est utilisé pour :

- Offrir une authentification unique (SSO) entre plusieurs services internes ou externes,
- Connecter une application à un annuaire LDAP d'entreprise,
- Renforcer la sécurité avec la double authentification (2FA),
- Créer un portail utilisateur pour accéder à différents outils,
- Gérer les rôles et les permissions depuis un seul endroit.

En résumé

Keycloak est une solution complète, moderne et open source pour gérer les utilisateurs, leurs connexions et leurs droits.
Il est particulièrement utile lorsqu'on souhaite centraliser l'accès à plusieurs services tout en gardant le contrôle sur la sécurité, la confidentialité et la flexibilité de déploiement.

Installation

Fichier docker-compose

Dans cette procédure, nous allons utiliser **Docker**. Une documentation préalable à ce sujet est disponible ici.
Commençons par créer un fichier `docker-compose.yaml`, adapté à nos besoins :

```
services:
  # Service de base de données PostgreSQL pour Keycloak
  keycloak-db:
    image: postgres:16                  # Utilise l'image officielle PostgreSQL version 16
    container_name: keycloak-db
    environment:
      POSTGRES_DB: keycloak             # Nom de la base de données
      POSTGRES_USER: keycloak           # Nom de l'utilisateur
      POSTGRES_PASSWORD: keycloak       # Mot de passe de l'utilisateur
    volumes:
      - keycloak-db-data:/var/lib/postgresql/data  # Persistance des données de la base
    networks:
      - proxy  # Connecté au réseau "proxy" (externe, voir plus bas)

  # Service Keycloak (serveur d'identité)
  keycloak:
    image: quay.io/keycloak/keycloak:24.0  # Image officielle Keycloak version 24.0
    container_name: keycloak
    command: >
      start
      --hostname=sso.domaine.com            # Nom d'hôte public visible par les utilisateurs
```

```
      --hostname-strict=false               # Autorise des hôtes alternatifs (utile avec des
reverse proxy)
      --proxy-headers=xforwarded            # Interprète les headers X-Forwarded-* du proxy
(option héritée)
    environment:
      KC_DB: postgres                       # Type de base de données
      KC_DB_URL: jdbc:postgresql://keycloak-db:5432/keycloak  # URL JDBC vers la base
PostgreSQL
      KC_DB_USERNAME: keycloak              # Identifiant pour se connecter à la BDD
      KC_DB_PASSWORD: keycloak              # Mot de passe pour se connecter à la BDD
      KC_HOSTNAME_STRICT: "false"           # (Redondant avec l'option de commande)
      KC_PROXY: "edge"                      # Mode proxy recommandé pour les reverse proxies
      KEYCLOAK_ADMIN: admin                 # Nom de l'administrateur initial
      KEYCLOAK_ADMIN_PASSWORD: admin        # Mot de passe de l'administrateur
    depends_on:
      - keycloak-db                         # Attend que la base de données soit disponible
    ports:
      - 8080:8080                            # Expose le port 8080 (interne) sur le port 800
(hôte)
    networks:
      - proxy                               # Connecté au réseau externe "proxy"
    restart: unless-stopped                 # Redémarre le service sauf si l’arrêt est manuel

# Volume Docker pour la persistance des données PostgreSQL
volumes:
  keycloak-db-data:

# Réseau Docker externe utilisé pour connecter ce service à un reverse proxy (comme Traefik ou
Apache/Nginx)
networks:
  proxy:
    external: true  # Le réseau "proxy" doit déjà exister (créé manuellement)
```

Démarrage

Une fois le fichier `docker-compose.yaml` créé, nous pouvons **lancer Keycloak** avec la commande suivante :

```
docker compose up -d
```

Configuration