Keycloak 이란?

- Redhat 에서 개발한 서비스를 대상으로 하는 인증 & 권한 부여 오픈소스
- single-sign-on이 가능한 오픈소스

- **OIDC**
  Oauth 2.0의 확장 인증 프로토콜로, **인증 (본인 증명)에 초점**을 맞춤

- **OAuth 2.0**
  권한 부여 프레임워크로, **데이터에 대한 액세스 권한 부여에 초점**을 맞춤
  OAuth 2.0 은 인증 프로토콜이 아님 (공식 홈페이지에 의거)

- **차이**
  OIDC는 기본적으로 OAuth 프로토콜을 기반으로 작동하는 기술로, 사용 목적에 큰 차이를 둠
  위에서 설명한대로, **OIDC는 인증** (사용자 개인정보가 담긴 id_token 등) /
  OAuth2.0 인가 (해당 플랫폼에 저장된 사용자 데이터 접근 Access_token)에 목적을 둠

- **KeyCloak OIDC 정리**
  KeyCloak은 기본 인증 방식이 OAuth 2.0을 베이스로 한 OIDC

  => KeyCloak 서버는 OAuth 2.0 권한 부여 프레임워크 위에 빌드된 OIDC를 사용하여
  인증을 가능하게 하는 사용하기 쉬운 권한 부여 서버를 제공하는 것 같다.

출처 : https://alice-secreta.tistory.com/28

Keycloak 관련 용어

- **OIDC**

: OAuth 가 권한 부여만 다루는 것이라면 OIDC 는 OAuth 를 포함하여 인증과 권한부여를 모두 포함한 것이다. SSO 의 구현을 위한 수단으로 사용된다.

- **Realm**

: 인증, 권한 부여가 적용되는 범위를 나타내는 단위이다. SSO 를 적용한다고 했을 때 해당 SSO 가 적용되는 범위는 Realm 단위이다.

- **Client**

: 인증, 권한 부여 행위를 대행하도록 맡길 어플리케이션을 나타내는 단위이다. 그 단위는 웹사이트 혹은 REST API 를 제공하는 서비스도 될 수 있다. 하나의 Realm 에 n개의 Client 를 생성, 관리할 수 있다.

- **User**

: Client 에 인증을 요청할 사용자를 나타낸다. 하나의 Realm 에는 Realm 에 종속된 n개의 User 를 생성하고 관리할 수 있다. 기본적으로 User 는 *Username, Email, FirstName, LastName* 으로 구성되어 있지만 Custom User Attribute 를 사용하면 사용자가 원하는 속성을 추가할 수 있다.

- **Role**

: User 에게 부여할 권한 내용을 나타낸다. 여기에는 Keycloak 의 REST API 를 사용할 권한을 부여할 수 있고 사용자가 정의한 권한을 부여할 수도 있다.