클라우드 네이티브 IAM(Identity and Access Management) 솔루션

박채림 [1], 전우재 [2], 박진형 [3], 박성훈 [4]
[1] 서울여자대학교 소프트웨어융합학과
[2] 금오공과대학교 컴퓨터공학과
[3] 부산대학교 IT 응용공학과
[4] 삼성 SDS

qkrdbsk28@naver.com, dnwo0719@kumoh.ac.kr, dyunames21@pusan.ac.kr, architectstory@gmail.com

Cloud Native IAM(Identity and Access Management) Solution

Chae-Rim Park[1], Woo-Jae Jeon[2], Jin-Hyung Park[3], Sung-Hun Park[3]
[1]Dept. of Software Convergence, Seoul Women's University
[2]Dept. of Computer Engineering, Kumoh University of Technology
[3]Dept. of IT Convergence and Application Engineering, Pusan University
[4]SamSung SDS

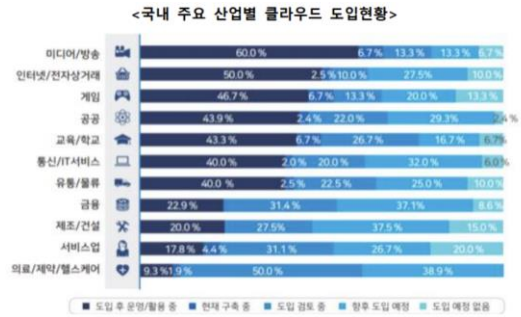
요 약

본 논문은 클라우드 환경에 적합한 IAM(Identity and Access Management) 솔루션을 제안한다. 오픈소스 라이브러리인 Keycloak[1]을 이용하여 그룹 별 권한 관리 및 권한에 따른 리소스 관리가 가능하도록 하며, 솔루션을 쉽게 도입하여 사용할 수 있도록 컨테이너 기술을 통해 신속하게 환경을 구축하고 배포할 수 있게 도와주는 플랫폼인 Docker 를 사용해 Docker image 형식으로 제공한다.

1. 서론

1.1. 제안 배경

Covid-19 이후 클라우드 컴퓨팅 시장은 급격한 성장세를 보이고 있으며 구글(Google), 마이크로소프트(MS), IBM, 아마존(Amazon) 등 대표적인 IT 기업들은 클라우드를 차세대 핵심 비즈니스로 꼽고 SaaS(Software as a Service), PaaS(Platform as a Service), IaaS(Infrastructure as a Service) 등 다양한 클라우드 컴퓨팅 서비스 및 제품들을 출시하고 있다.



(그림 1) 국내 주요 산업별 클라우드 도입 현황 [2]

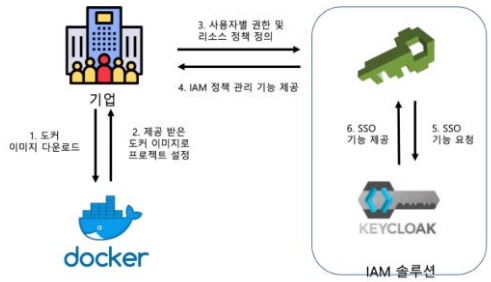
또한 정부에서도 이러한 변화에 발 맞추어 국가기관 및 공공기관에게 클라우드 환경을 적극적으로 도입할 것을 법률[3]을 통해 권장하고 있다.

그러나 클라우드에 대한 많은 관심과 함께 클라우드 보안 위협 또한 증가하고 있다.[4] 클라우드를 사용하는 서비스에서는 늘 보안 위협이 뒤따르는데 그중 제일 문제가 되는 것이 바로 데이터 유출이다. 이는 불충분한 인증 및 접근 관리로 인해 발생하며 클라우드를 이용하는 고객에게 큰 위험 요소가 된다.

이러한 문제를 해결하고자 종합적으로 클라우드 컴퓨팅의 리소스를 제어할 수 있는 인증과 접근 관리 체계를 바탕으로 한 IAM 솔루션을 제안하게 되었다.

2. 본론

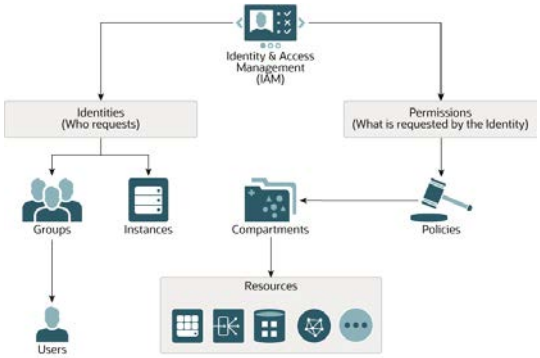
2.1. 솔루션 모델



(그림 2) IAM 솔루션 모델

그림 2는 본 논문에서 제안하고자 하는 IAM 솔루션 모델이다. IAM 솔루션 적용이 필요한 기업에서는 본 논문의 IAM 솔루션 Docker Image를 다운받아 기업의 프로젝트에 적용한 뒤, IAM 솔루션에서 제공하는 여러 기능들을 사용하여 사용자 그룹별 권한 및 리소스 관리 정책을 정의할 수 있다. 또한 제안하는 IAM 솔루션은 클라우드와 연동되어 있기 때문에 클라우드 환경에서 사용하기 적합한 솔루션이다. SSO(Single-Sign-On) 기능이 필요한 경우에는 오픈소스 라이브러리인 Keycloack을 통해 SSO 기능을 제공받아 사용할 수 있다.

본 논문의 IAM 솔루션에서는 액세스하는 모든 요청에 대해 그림 3과 같이 정의한 Keycloack의 IAM 모델 구조에 따라 인증을 진행한다.



(그림 3) IAM 모델 구조 [5]

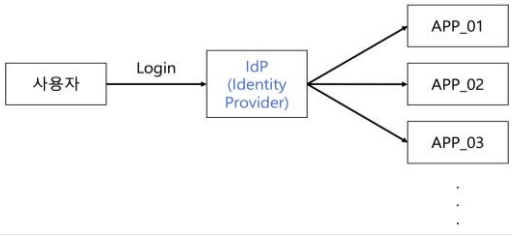
기업에서 정의한 사용자 그룹별 권한 및 리소스 관리 정책을 DB(DataBase)에 저장하며 요청에 따라 사용자의 정보를 전송하여 비교할 수 있도록 한다.

2.2. 주요 기능

본 논문에서 제안하는 IAM 애플리케이션의 주요 기능은 다음과 같다.

2.2.1. 통합 인증 - SSO(Single-Sign-On)

해당 솔루션에서는 하나의 사용자 정보를 이용하여 연결된 다수의 애플리케이션에 접근이 가능하도록 하는 SSO 기능을 제공한다.



(그림 4) SSO 기능

SSO 기능을 통해 중앙 관리자가 수월하게 관리 대상(사용자 및 고객)을 관리할 수 있도록 하며, 해당 기능은 오픈소스 라이브러리인 Keycloak을 이용한다.

2.2.2. 사용자 그룹화 관리

해당 솔루션에서는 사용자들을 그룹별로 나누어 관리할 수 있는 기능을 제공한다. 그룹과 해당 그룹에 추가를 원하는 사용자를 지정해주면 그룹별로 묶어서 사용자 관리가 가능하다.

2.2.3. 사용자 및 그룹 권한 관리

해당 솔루션에서는 사용자 및 그룹이 가질 수 있는 권한에 대한 관리 기능을 제공한다.

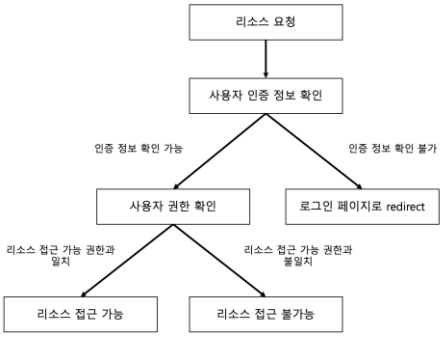
```
{
  Groups: ["network-group"],
  Roles: ["network-management.viewer"]
}

{
  Groups: ["network-group"],
  Users: ["user_1", "user_2"],
  Roles: ["network-management.*"]
}
```

(그림 5) JSON 형식으로 정의한 정책

JSON 형식의 정책을 사용하여 그룹 및 사용자에게 부여하고자 하는 권한을 정의할 수 있다. 그림 5와 같이 그룹에 속한 사용자 전체에게 동일한 권한을 부여할 수 있으며, 특정 사용자에게 특정 권한을 부여하는 것 또한 가능하다.

2.2.4 리소스 접근 권한 제어

해당 솔루션에서는 2.2.3의 권한 관리 기능을 통해 정의한 권한에 따라 리소스 접근 제어 기능을 제공한다.



(그림 6) 리소스 접근 제어 과정

그림 6 은 리소스 접근 제어 과정을 설명하는 그림이다. 리소스에 접근하려는 사용자가 URL 을 통해 리소스 요청을 하면 사용자 인증의 여부를 확인한다. 이때 이미 로그인을 하여 인증 정보가 확인되는 상황에는 바로 사용자의 권한과 리소스 접근 가능 권한을 비교하여 리소스를 제공 여부를 결정한다. 요청 당시 사용자 인증 정보를 확인할 수 없는 경우에는 로그인 페이지로 이동하게 되며 로그인 이후에 사용자 권한을 확인하여 리소스에 대한 접근 제어 동작을 수행한다.

3. 결론

3.1. 솔루션 적용 결과

솔루션 적용 예시 프로젝트는 기업의 입장에서 설계되었으며 사용자는 크게 그룹 관리자와 그룹 내 사용자로 구분된다. 그룹 관리자는 소속된 그룹의 사용자들을 관리할 수 있는 권한을 가지고 있는 사용자를 뜻한다. 그룹 관리자 계정으로 로그인할 경우 그림 7 과 같은 사용자 추가 페이지에서 추가하고자 하는 사용자의 이메일과 이름을 입력하여 본인 그룹에 추가할 수 있다.

(그림 7) 사용자 추가 페이지

그룹 관리자에 의해 추가된 사용자는 자동 생성된 임시 비밀번호를 받게 되며 이후 변경이 가능하다. 권한에 따른 접근 제어 기능을 통해 사용자 추가 페이지는 그룹 관리자 권한을 가진 사용자만이 접근 가능하다는 것을 확인할 수 있었다.

3.2. 차별점 및 주요 이점

기존의 어플리케이션의 경우 Spring MVC(Model-View-Controller) 디자인 패턴을 기반으로 구조화되어 있기 때문에 Spring 의 WebMvcConfigurer 를 상속받아 view 를 추가하여 이름을 지정한다. 해당 방식은 보안이 적용되지 않기 때문에 외부에서 누구나 원하는 view 에 접근이 가능하다. Spring 에서 제공하는 보안 프레임워크인 SpringSecurity 의 SecurityFilterChain 을 사용하면 URL 별 접근에 보안이 적용되어 로그인과 같이 사용자의 권한을 확인할 수 있는 View 로 redirect 된다. 그 후 사용자 정보를 받아 사용자가 가진 권한(Role)을 확인하고 그에 맞는 리소스 접근 제어 기능을 제공한다.[6] 그러나 해당 방법은 애플리케이션마다 새로운 Config 파일을 작성하여 적용해주어야 한다는 번거로움이 따른다.

본 논문에서는 오픈 소스 라이브러리인 Keycloak 를 사용하여 URL 에 따른 접근을 보호하는 것 뿐만 아니라 Keycloak 자체에서 애플리케이션과 서비스를 IAM 솔루션을 사용하여 보호할 수 있다. IAM 솔루션은 애플리케이션에 대한 인증 뿐만 아니라 사용자나 리소스의 액세스 수준, 역할, 권한을 관리하며 이러한 특징은 마이크로 서비스 환경에서도 적합하다. 애플리케이션에 접근하려고 하면 독립적인 Keycloak 인증 서버로 redirect 하며 이 Keycloak 서버는 SSO(Single-Sign-On)에서 Open ID Connect, OAuth 2.0, SAML 같은 개방형 프로토콜을 사용하여 [7] 사용자가 애플리케이션 단위(또는 Tenant 단위)로 격리하고 애플리케이션에서도 사용자에 대한 정보가 토큰화해 사용자에 대한 정보를 보호할 수 있다.

3.3. 활용방안

본 논문에서 제안하는 IAM 솔루션은 클라우드 환경에서 사용자 및 권한 관리와 리소스 제어가 필요한 다양한 분야의 서비스에서 사용될 수 있을 것이라 기대한다.

* 본 프로젝트는 과학기술정보통신부 정보통신창의 인재양성사업의 지원을 통해 수행한 ICT 멘토링 프로젝트 결과물입니다.

참고문헌

[1] Keycloak 공식 홈페이지, Single-Sign-On, https://www.keycloak.org/
[2] 조성현, 인공지능과 언택트 시대, 국내 주요산업의 클라우드 도입 현황 및 전망, 정보통신산업진흥원, 2020-제 01 호, 4 페이지
[3] 클라우드컴퓨팅 발전 및 이용자 보호에 관한 법률 제 12 조, 제 20 조 (2020. 12. 10 시행)
[4] 오민석, 클라우드 서비스의 보안 취약점과 대응방안, 춘계학술발표대회, 한국정보처리학회, 2019 년, 3 페이지.
[5] Oracle Cloud Infrastructure, https://docs.oracle.com/ko/solutions/oci-security-checklist/authorization1.html#GUID-F63D3D94-2725-44DE-B7D6-2A884A48DBA0
[6] Spring Guide, https://spring.io/guides/gs/securing-web/
[7] Keycloak 공식 홈페이지, Standard Protocols, https://www.keycloak.org/