# 신뢰기반 디지털 ID 관리기술

## - 아이디제공자의 구축과 관리 -

한국과학기술정보연구원 조진용
jiny92@kisti.re.kr

2024년 8월 29일(목)

KAFE

# Audience

## 대상

¶ 아이디제공자를 구축하고자 하는 학연 기관 관계자

## 수준

¶ 초급+α
¶ IT 관련 기초지식 및 PHP 문법 이해 필요

## 참고

¶ 강의자료: https://edu.kafe.or.kr

KAFE

# Agenda

1교시 (9:30AM - 10:20AM)

- ¶ 기초 지식

2교시 (10:30AM – 11:20AM)

- ¶ 아이디제공자의 구축

3교시 (11:30AM - 12:20AM)

- ¶ IdP의 검증/관리와 KAFE 가입

KAFE

# 1교시: 기초 지식

KAFE

# Federated Identity Management (FIM)

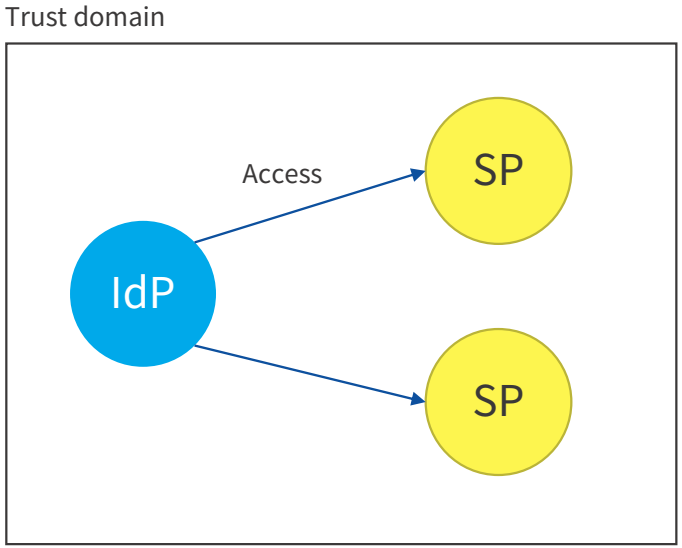
- 정의
  - 하나의 디지털 ID를 사용하여;
    서로 다른 도메인 내 존재하는 시스템들에 접근케 해주는;
    인증 및 인가 프레임워크

- 특징
  - 통합인증
  - 신뢰관계
  - 분산관리

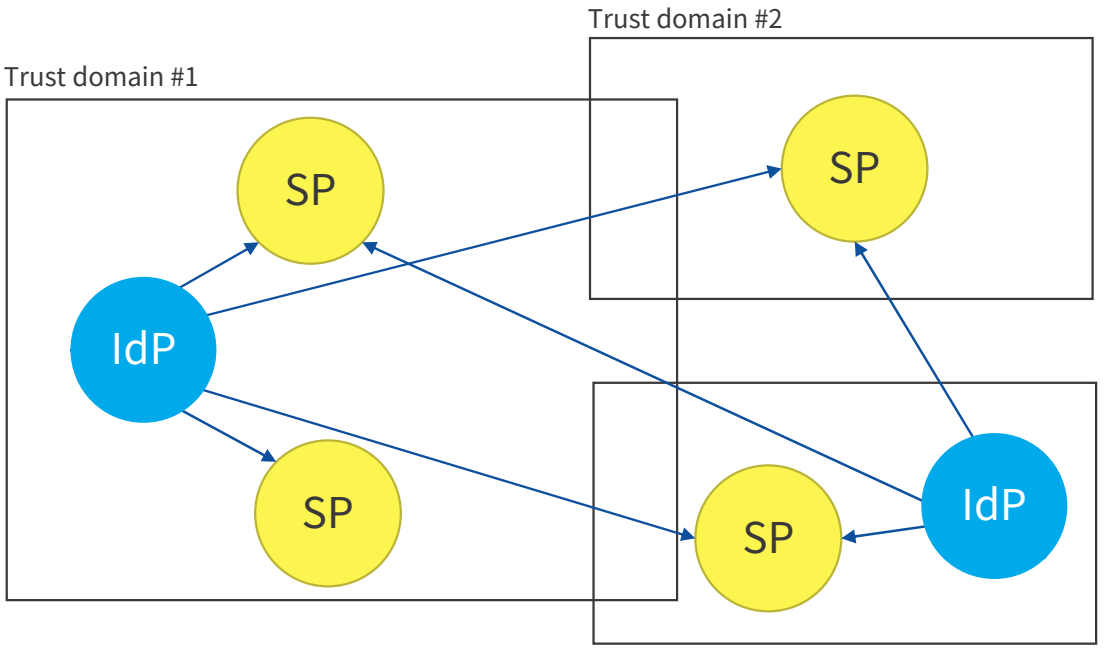
- 작동원리
  - 사용자는 한 시스템(Identity provider, IdP)에서 인증
  - 인증된 사용자 정보는 다른 신뢰할 수 있는 시스템(Service provider, SP)와 공유
  - SP는 인증된 사용자 정보를 신뢰하고 사용자에게 접근 권한을 부여

# 멀티도메인 접근

- 인프라 간 상호 운용성 및 접근성 확대
- 안전한 기관 외부 서비스제공자 접근 가능 (클라우드 서비스 등)




SSO(통합인증) in a single domain (e.g., a university)
Centralized identity management




SSO(통합인증) in multiple domains (e.g., universities)
Federated identity model

KAFE

# FIM의 장점

- 사용자 편의성
  - 여러 계정을 관리할 필요가 없음
- 보안 강화
  - 인증 관리의 중앙화로 기관 외부 서비스에 대한 보안 정책의 일관성 확보
- 관리 효율성
  - 사용자 계정 관리와 접근제어의 간소화
- 비용 절감
  - 여러 시스템에서 중복된 ID 관리 작업을 줄임

KAFE

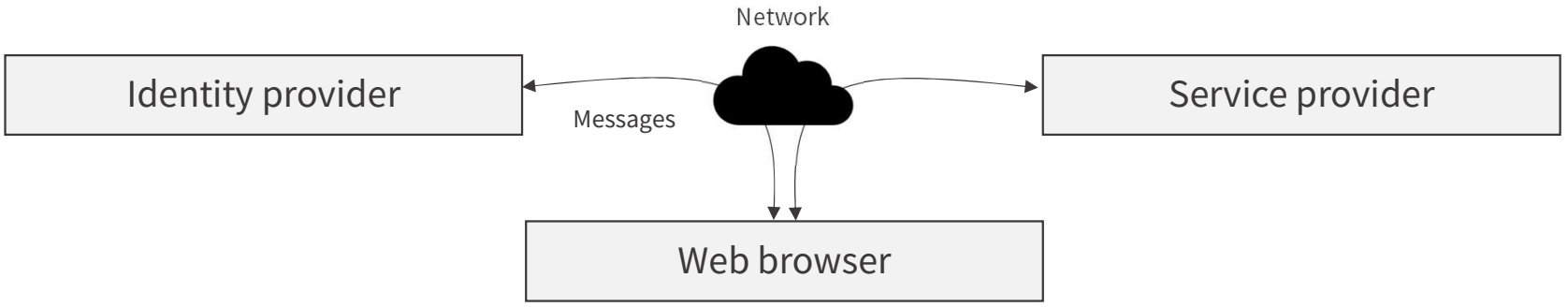
# 공개 표준

- SAML (Security Assertion Markup Language)
- OIDC/OAuth2 (OpenID Connect)

| SAML | OIDC | OAuth2 |
|---|---|---|
| Developed by OASIS in 2001 | Developed by the OpenID Foundation in 2014 | Developed by Twitter and Google in 2006 |
| Primary for authentication | Authentication & Authorization | Authorization |
| XML-based assertions | JSON Web Tokens (JWTS) | JSON Web Tokens (JWTS) |
| HTTP redirects, GET, POST | Authorization code flow, ID token for authentication | Grant types: authorization code, implicit, password credential, client credentials |
| Browser, service provider, identity provider | Browser, OIDC provider, OIDC client | Browser, OAuth2 provider, OAuth2 client |
| Enterprise for web SSO | Web and Mobile SSO | Social SSO |

# FIM의 개체

| Identity provider (IdP) | Service provider (SP) |
|---|---|
| OIDC provider | OIDC client |
| Manage user identities and associated attributes | Offers services or resources relying on IdP's identity information (a.k.a., relying party) |
| Providing identity information to service providers | Trust the IdP to authenticate users |
| Verifies the user's credentials and issues an assertion or token | Redirects a user to the IdP for authentication.<br>Receives assertion or token and grant access to the user |
| e.g., social login provider (Google, Facebook, etc.) | Cloud services |



# 데모 I

Inter-federation (Global FIM)

https://ieeexplore.ieee.org/Xplore/home.jsp

- Identity provider: KISTI (KR)
- Service provider: IEEExplore (US)
- Standard: SAML

# 데모 II

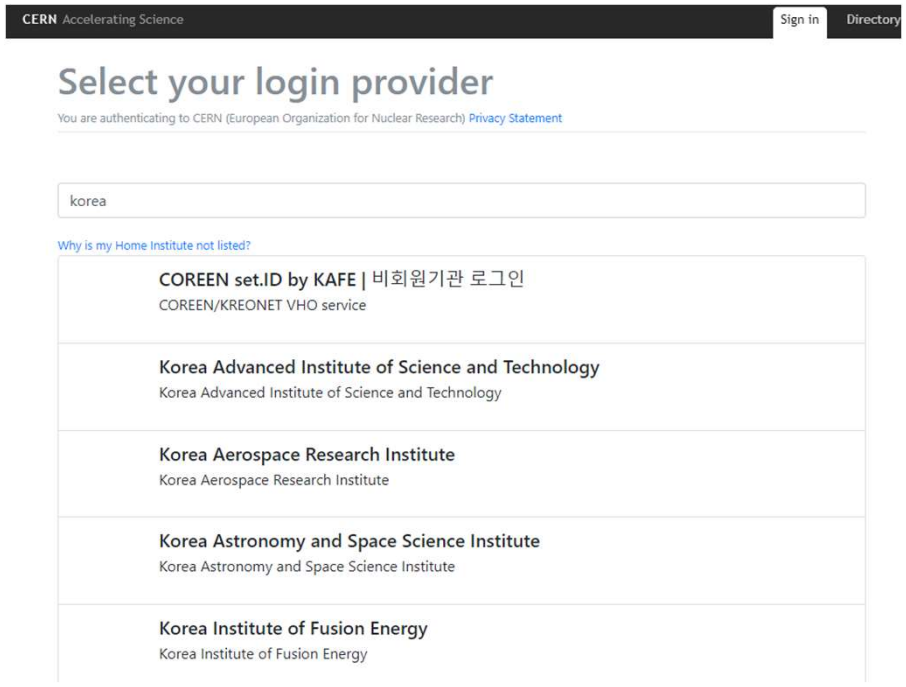
Local-federation (Domestic FIM)

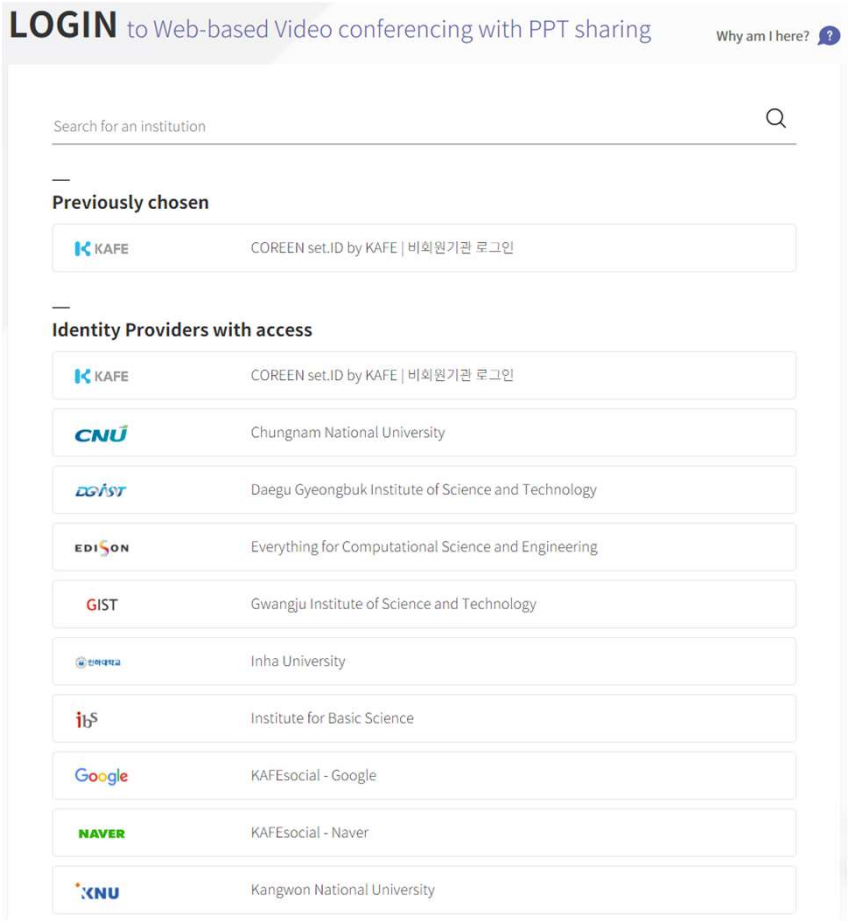
https://webinar.kafe.or.kr

- Identity provider: KISTI (KR)
- Service provider: Webinar(KR)
- Standard: SAML

# 탐색서비스

- Central Discovery Service
  - FIM 운영자가 제공
- Embedded Discovery Service
  - 서비스제공자가 제공

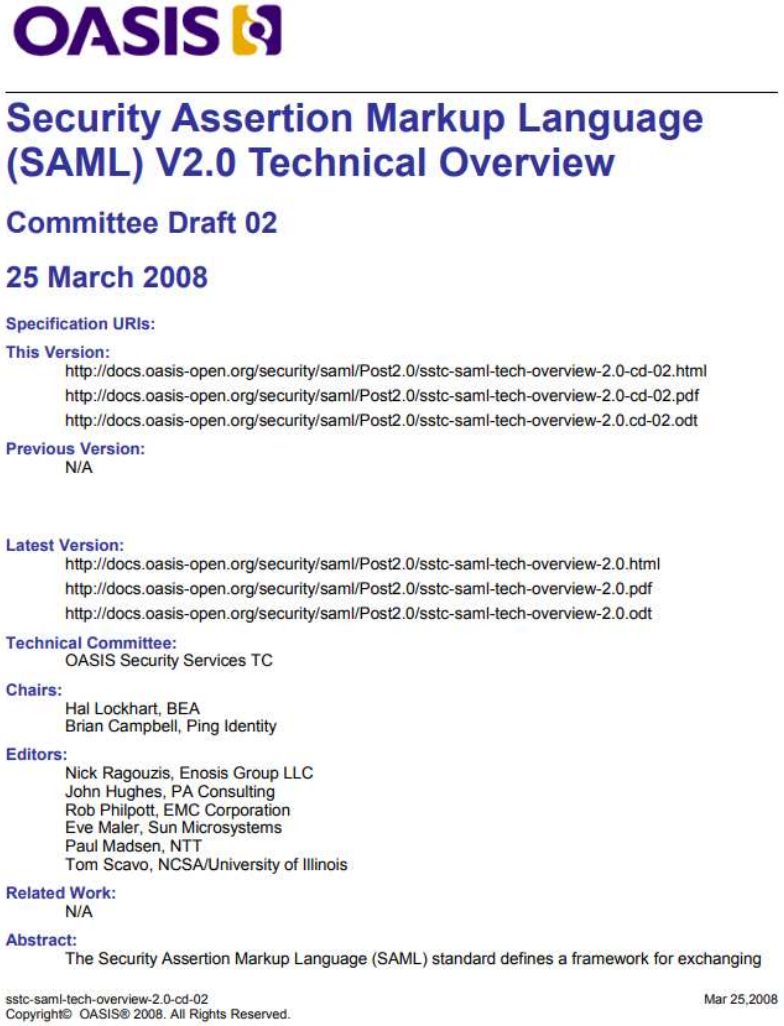


# SAML[샘엘]

SAML (Security Assertion Markup Language)
웹 통합인증(Single Sign On)을 위한 국제 공개표준
Federated authentication에 사용되는 주요 규약

- SAML 1.1(`03), SAML2.0(`05, Errata(정오표)/`19)
  - OASIS 보안서비스 기술위원회 제정
- 연구교육 분야 활용(`05~)
  - SWITCH(스위스 NREN)
- 강화된 보안 vs 적용 어려움
- 웹 환경 지원(모바일 환경 미지원)
  - OAuth2/OIDC 출현



OASIS

**Security Assertion Markup Language (SAML) V2.0 Technical Overview**

**Committee Draft 02**

**25 March 2008**

**Specification URIs:**

**This Version:**
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0-cd-02.html
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0-cd-02.pdf
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.cd-02.odt

**Previous Version:**
N/A

**Latest Version:**
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.pdf
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.odt

**Technical Committee:**
OASIS Security Services TC

**Chairs:**
Hal Lockhart, BEA
Brian Campbell, Ping Identity

**Editors:**
Nick Ragouzis, Enosis Group LLC
John Hughes, PA Consulting
Rob Philpott, EMC Corporation
Eve Maler, Sun Microsystems
Paul Madsen, NTT
Tom Scavo, NCSA/University of Illinois

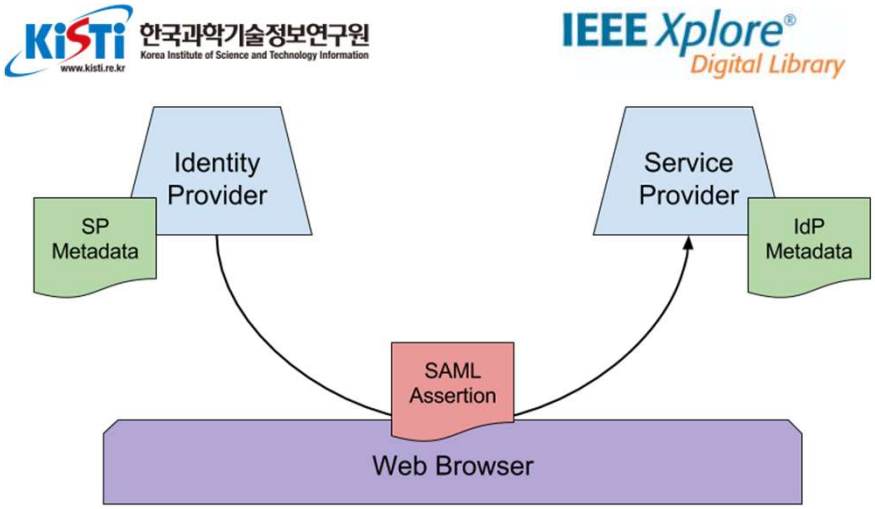
**Related Work:**
N/A

**Abstract:**
The Security Assertion Markup Language (SAML) standard defines a framework for exchanging

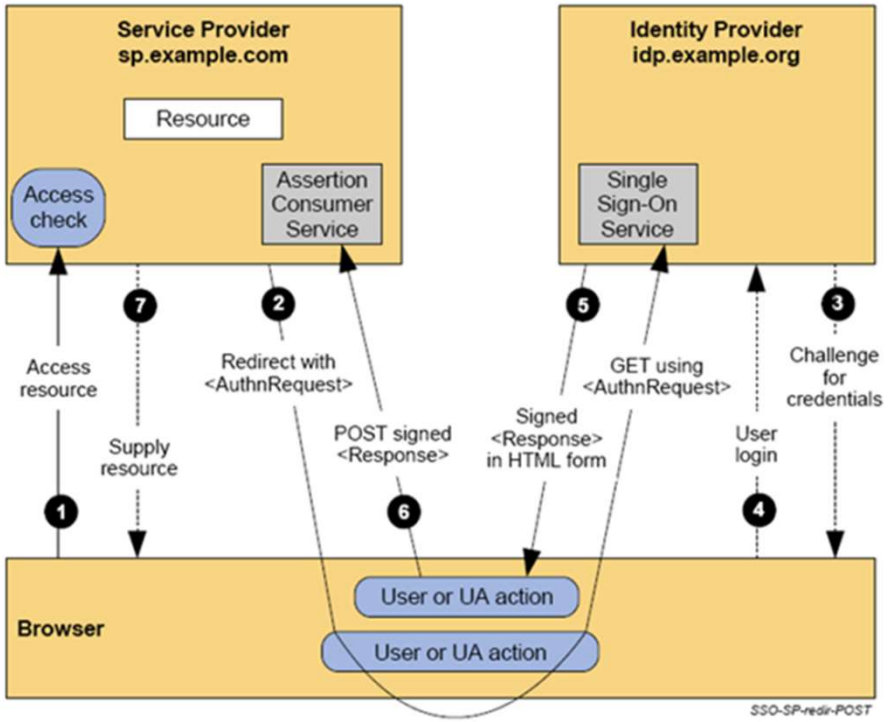
sstc-saml-tech-overview-2.0-cd-02 Mar 25,2008

# Web SSO profile

- IDP와 SP는 직접 통신하지 않음(사용자 브라우저가 정보 전달)
- SP는 IdP의 계정 DB에 접근할 수 없음
- 메타데이터를 교환하지 않으면 통신 불가능




(1) SAML 메타데이터 교환



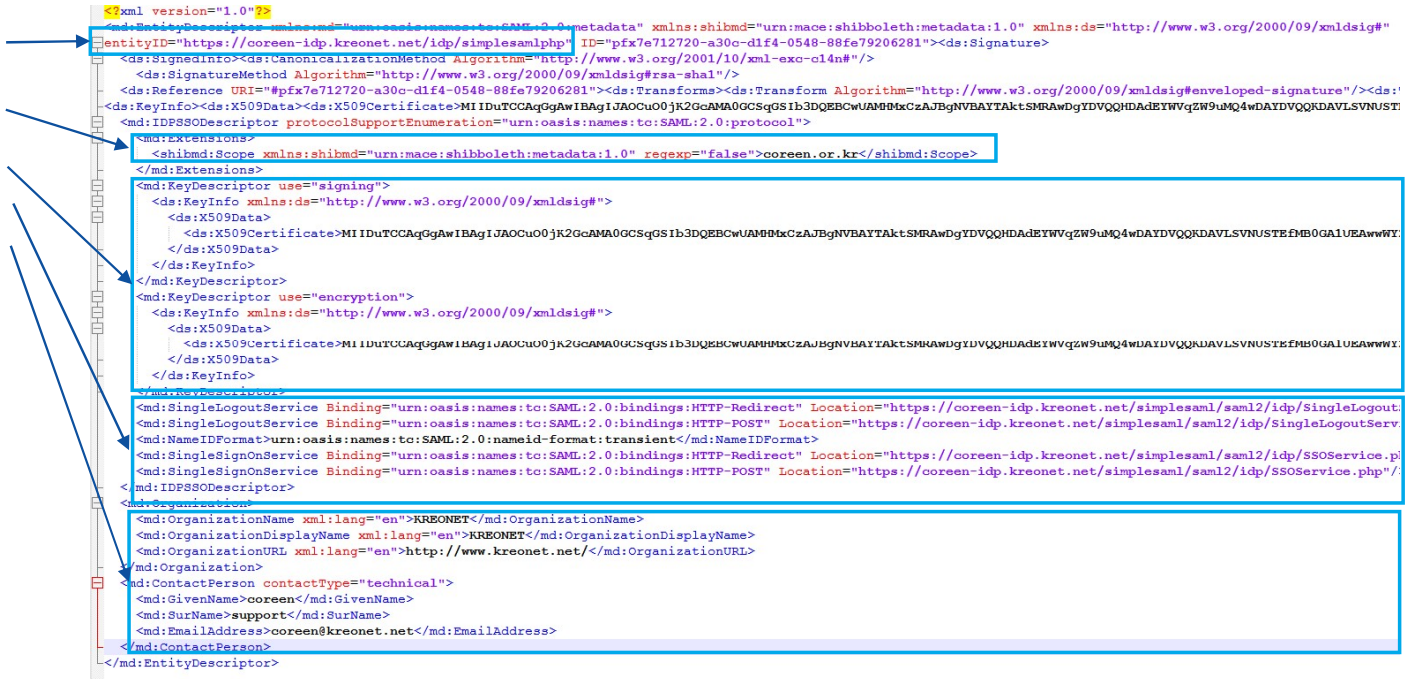

(2) 인증메시지 교환

Source: Wikipedia, ECLIPSE Foundation

# 메타데이터

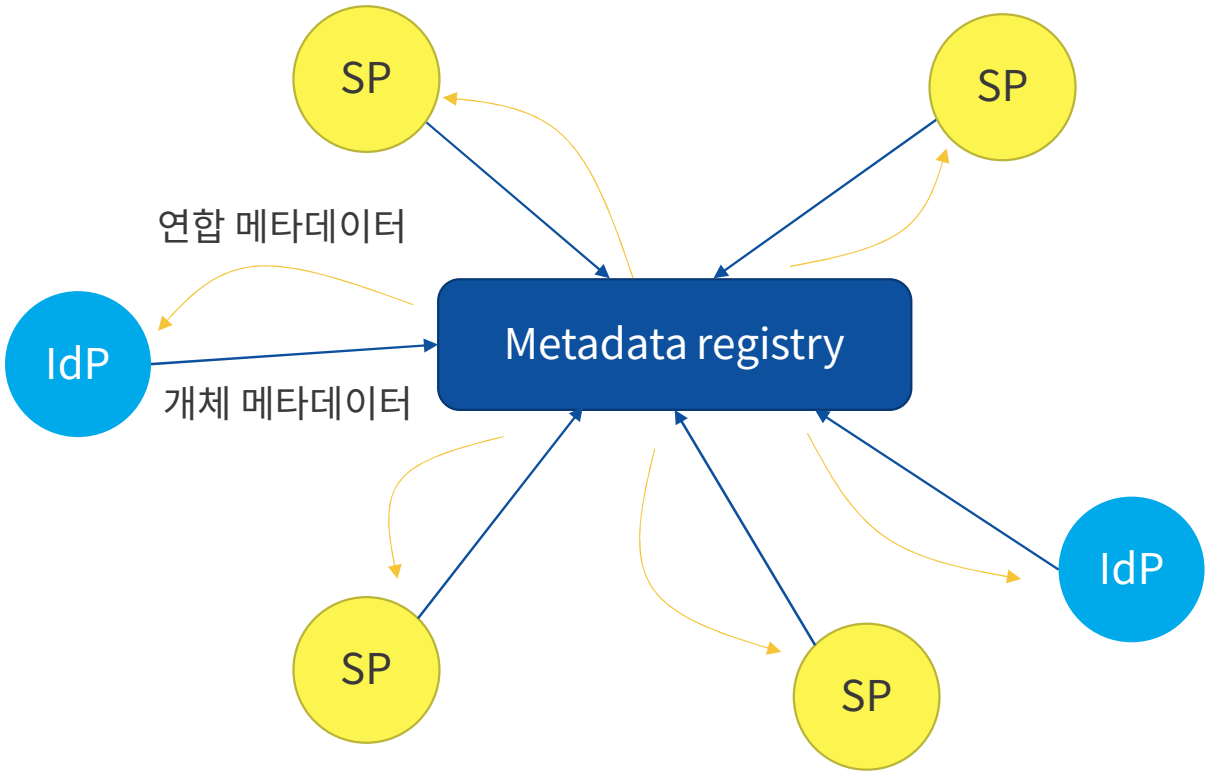
- Exchange public keys for encryption and signature
- Exchange endpoint URLs

- ✓ 개체식별자(EntityID)
- ✓ 기관도메인(Scope)
- ✓ 공개키(Signature/encryption)
- ✓ 서비스 주소(Protocol endpoints)
- ✓ Contacts



# 메타데이터의 교환

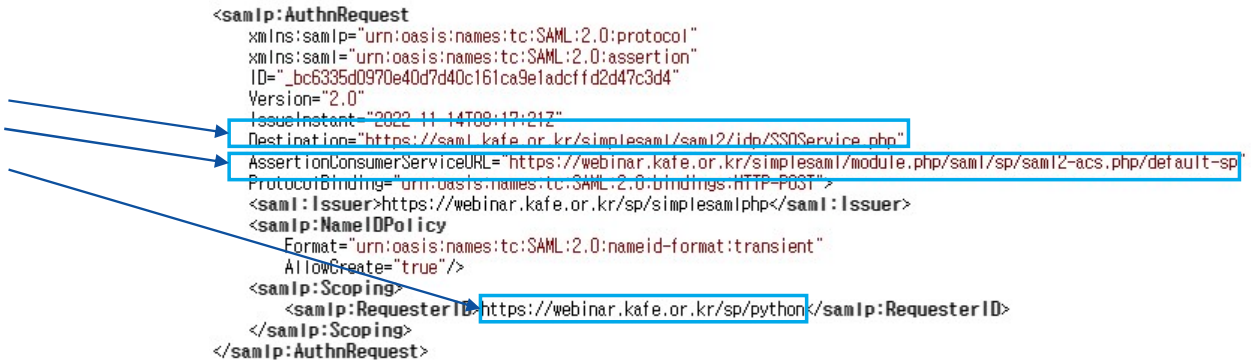
- FIM Operator가 개체 메타데이터를 수집 후 연합 메타데이터를 배포
  - 연합 메타데이터



# SAML 인증요청 메시지

- 서비스제공자가 인증요청 메시지를 생성

✓ IdP의 SSO URL
✓ ACS(Assertion Consumer Service) 주소
✓ 서비스제공자 식별자



# SAML 인증응답 메시지

- IdP가 인증응답 메시지를 생성
- Assertion: 인증에 성공한 사용자의 신원정보와 인증정보를 담고 있는 XML statement

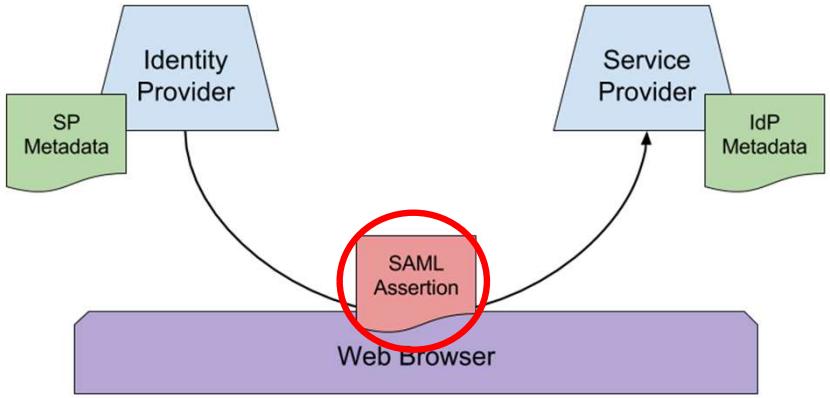
- ✓ 메시지 발행자(Issuer)
- ✓ Signature
  Digest를 개인키로 암호화
- ✓ X.509 인증서
- ✓ 인증정보: 누가, 언제, 어떤 서비스제공자를 위해;
  어떤 인증방식(예, 비밀번호)으로 로그인 했고;
  메시지가 언제까지 유효한가?
- ✓ 사용자 속성값

```
<samlp:ArtifactResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                        ID="s2d9926a42ce0f17629c6af30a6dd8a15fa7409415" InResponseTo="RkN2Yn1EM"
                        Version="2.0" IssueInstant="2007-01-02T20:48:35Z">
    <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
        idp.ssocircle.com
    </saml:Issuer>
    <samlp:Status xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
        <samlp:StatusCode  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
            Value="urn:oasis:names:tc:SAML:2.0:status:Success">
        </samlp:StatusCode>
    </samlp:Status>
    <samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                        ID="s276a3300f0e2b3d7f67a800d332156f0fdb99c5d0"
                        InResponseTo="NKRDzVK7e" Version="2.0" IssueInstant="2007-01-02T20:48:35Z">
        <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
            idp.ssocircle.com
        </saml:Issuer>
        <samlp:Status xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
            <samlp:StatusCode  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                Value="urn:oasis:names:tc:SAML:2.0:status:Success">
            </samlp:StatusCode>
        </samlp:Status>
        <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Version="2.0"
                         ID="s25d1bbb23ed13c6da325a6f7637c6dc5d9e7f13a9" IssueInstant="2007-01-02T20:48:35Z">
            <saml:Issuer>
                idp.ssocircle.com
            </saml:Issuer>
            <saml:Subject>
                <saml:NameID NameQualifier="idp.ssocircle.com" Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">
                    amv9OywHQrEW2rmaRViMpWX4qyDA
                </saml:NameID>
                <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                    <saml:SubjectConfirmationData NotOnOrAfter="2007-01-02T20:58:35Z" InResponseTo="NKRDzVK7e"
                        Recipient="http://cgi.cohos.de:80/cgi-bin/zxid" ></saml:SubjectConfirmationData>
                </saml:SubjectConfirmation>
            </saml:Subject>
            <saml:Conditions NotBefore="2007-01-02T20:48:35Z" NotOnOrAfter="2007-01-02T20:58:35Z">
                <saml:AudienceRestriction>
                    <saml:Audience>
                        http://cgi.cohos.de:80/cgi-bin/zxid?o=B
                    </saml:Audience>
                </saml:AudienceRestriction>
            </saml:Conditions>
            <saml:AuthnStatement AuthnInstant="2007-01-02T20:48:35Z" SessionIndex="s24bfc21323ee9c117bf5769a074be1ff177262701">
                <saml:AuthnContext>
                    <saml:AuthnContextClassRef>
                        urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
                    </saml:AuthnContextClassRef>
                </saml:AuthnContext>
            </saml:AuthnStatement>
        </saml:Assertion>
    </samlp:Response>
</samlp:ArtifactResponse>
```

# 속성(Attributes)

Assertion에 속성(Attributes) 포함

Friendly name/OID 형식 표기
¶ Friendly name
(예): cn/commonName, sn/surName,
¶ OID(Object Identifier)
(예): URN:OID:2.5.4.3



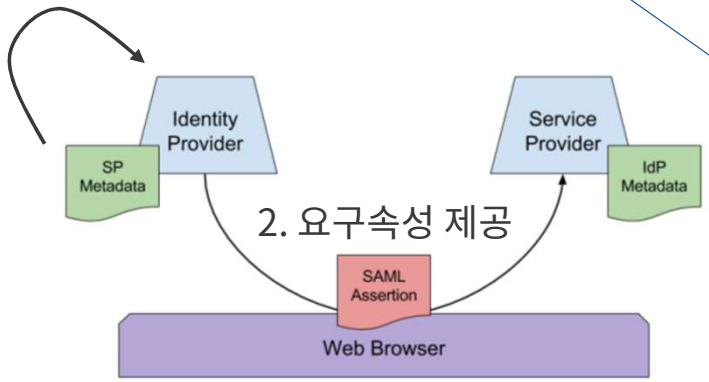

displayName
- SAML Name: urn:oid:2.16.840.1.113730.3.1.241
- LDAP source attribute: suDisplayname
- Example: Prof. John Doe

```
<saml:AttributeStatement>
    <saml:Attribute
        Name="urn:oid:2.16.840.1.113730.3.1.241"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml:AttributeValue
            xsi:type="xs:string">Jinyong JO
        </saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
        Name="urn:oid:0.9.2342.19200300.100.1.3"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml:AttributeValue
            xsi:type="xs:string">jinyong.jo@gmail.com
        </saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
        Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml:AttributeValue
            xsi:type="xs:string">staff
        </saml:AttributeValue>
        <saml:AttributeValue
            xsi:type="xs:string">member
        </saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
        Name="urn:oid:2.5.4.10"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml:AttributeValue
            xsi:type="xs:string">KISTI
        </saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
        Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml:AttributeValue
            xsi:type="xs:string">jiny92@coreen.or.kr
        </saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
        Name="urn:oid:1.3.6.1.4.1.25178.1.2.9"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml:AttributeValue
            xsi:type="xs:string">coreen.or.kr
        </saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
        Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml:AttributeValue>
            <saml:NameID
                NameQualifier="https://coreen-idp.kreonet.net/idp/simplesamlphp"
                SPNameQualifier="https://filesender.kreonet.net/sp/simplesamlphp"
                Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">d74f17c949df30493dc8cf7171959c01716f1d03
            </saml:NameID>
        </saml:AttributeValue>
    </saml:Attribute>
</saml:AttributeStatement>
```

# 전달되는 속성값

- 서비스제공자는 SP 메타데이터에 요구속성을 명시
- IdP는 SP 메타데이터를 확인하고 요구속성 중 전달 가능한 속성을 전달

1. 서비스제공자의 요구속성 확인




```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:md="urn:oasis:n
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:saml="urn:oasis:names:tc:SAML:2.0:a
xmlns:shibmd="urn:mace:shibboleth:metadata:1.0" xmlns:mdui="urn:oasis:names:tc:SAML:met
xmlns:mdattr="urn:oasis:names:tc:SAML:metadata:attribute" xmlns:mdrpi="urn:oasis:names:
xmlns:idpdisc="urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-protocol" xmlns:init=
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xi="http://www.w3.org/2001/
entityID="https://snu.bookcube.biz/sp/shibboleth">
  <md:Extensions>
   <mdrpi:RegistrationInfo registrationAuthority="http://kafe.kreonet.net" registrationI
  </md:Extensions>
```

...,

```
<md:AttributeConsumingService index="0">
 <md:ServiceName xml:lang="en">Bookcube for Seoul National University</md:ServiceName>
 <md:ServiceDescription xml:lang="en">It is a service that allows you to use e-books online. Users can
 <md:RequestedAttribute FriendlyName="eduPersonPrincipalName" Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" Na
 <md:RequestedAttribute FriendlyName="eduPersonAffiliation" Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1" Name
 <md:RequestedAttribute FriendlyName="email" Name="urn:oid:0.9.2342.19200300.100.1.3" NameFormat="urn:o
 <md:RequestedAttribute FriendlyName="displayName" Name="urn:oid:2.16.840.1.113730.3.1.241" NameFormat="
</md:AttributeConsumingService>
```

SAML SP Metadata

# 주요 속성

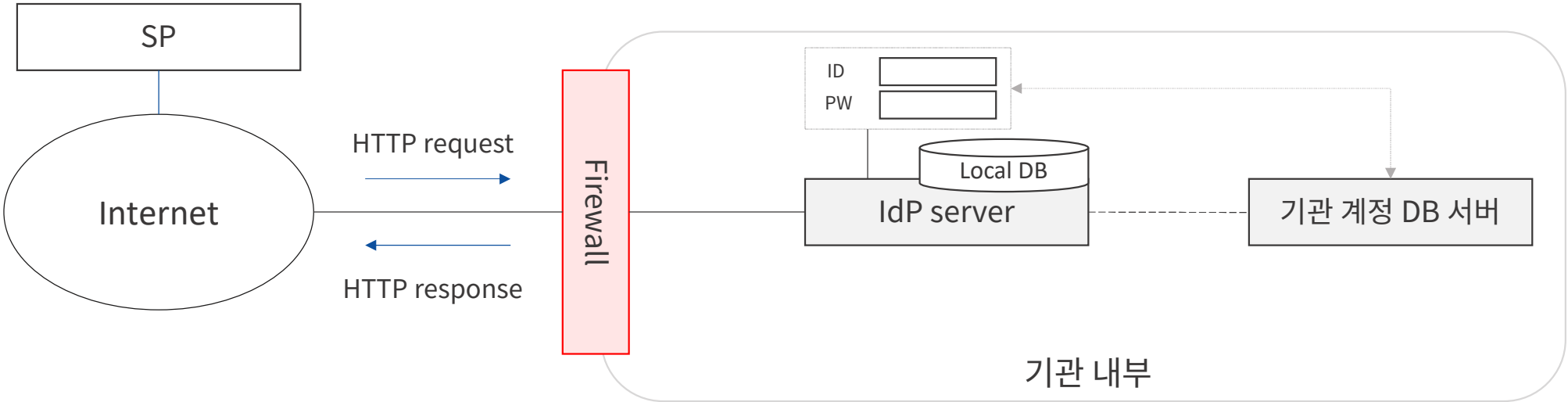
- https://www.kafe.or.kr/attributeMap 참조 (KAFE 표기법)
  - 속성은 OID 표기법을 따름
    - 예, mail == urn:oid:0.9.2342.19200300.100.1.3

| 속성명 | 예시 | 설명 |
|---|---|---|
| eduPersonTargetedID | Fdeef0012abcf22af | 특정 서비스 한정 (사용자) 고유식별자 |
| eduPersonPrincipalName | fedff0323aa@kafe.or.kr | 글로벌 (사용자) 고유식별자<br>• 전자우편 주소와 다름 |
| eduPersonScopedAffiliation | faculty@kafe.or.kr | 기관 내 직무정보<br>• 전자우편 주소와 다름 |
| eduPersonEntitlement | storage.read | 리소스 접근 자격 |
| displayName | John Doe | 성명 |
| cn (commonName) | John Doe | 성명 |
| Mail | john@kafe.or.kr | 전자우편<br>• 'email' 속성은 사용하지 않음 |

# 2교시: 아이디제공자의 구축

# IdP의 역할 및 구동 환경

- 인증 요청의 수신
- ID/PW의 입력 (사용자)
- ID/PW의 유효성 검증 (with 사용자 계정 DB)
- 사용자 정보의 획득 (from 사용자 계정 DB)
- 사용자 정보의 가공 (속성화)
- 접근 제어 및 접근 기록 관리, etc.
- 사용자 동의
- 인증 응답의 송신



# IdP 서버의 설치 환경

- Rocky 리눅스 8.8 (권장) 서버
  - 서버 설치 후 Clean slate 유지

- 서버 시스템 요구 사양 (최소)
  - 2 vCPU
  - 16 GB Memory
  - 100Mbps 네트워크 카드
  - 32 GB HDD

- 도메인 이름 및 공인 IP 할당

- 소프트웨어 방화벽 비활성화 (필요 시 KAFE 소프트웨어 설치 후 활성화)
  - iptables, selinux, firewalld 등

KAFE

# IdP 서버의 운영 환경

- Apache 2.4 이상
  - HTTPS용 TLS 인증서 필요 (기관보유 wildcard 인증서)
- PHP 8.2
- DBMS 클라이언트 드라이버 또는 RESTful API (사용자 계정 DB 연동 목적)
  - AD, LDAP, MSSQL, Tibero, MySQL, Oracle, etc.
- NTP 시간 동기화
  - 기관 NTP 서버 연동 또는 time.kriss.re.kr (default) 이용
- 방화벽
  - 인터넷 ~ IdP 서버 구간: TCP 80/443 입출력 개방
  - IdP 서버 ~ 계정 DB 서버 구간: DBMS 포트 개방

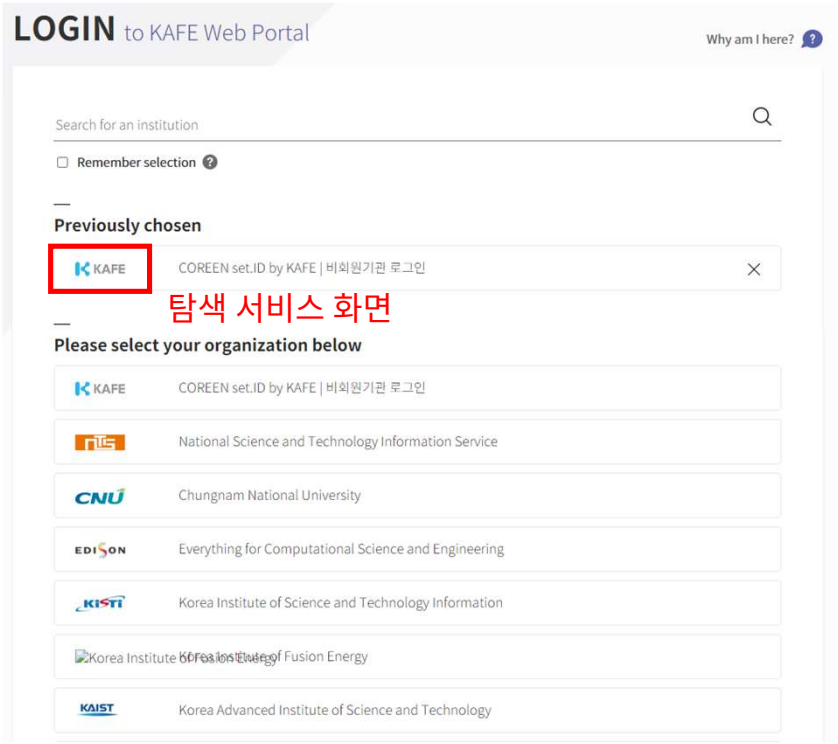
KAFE

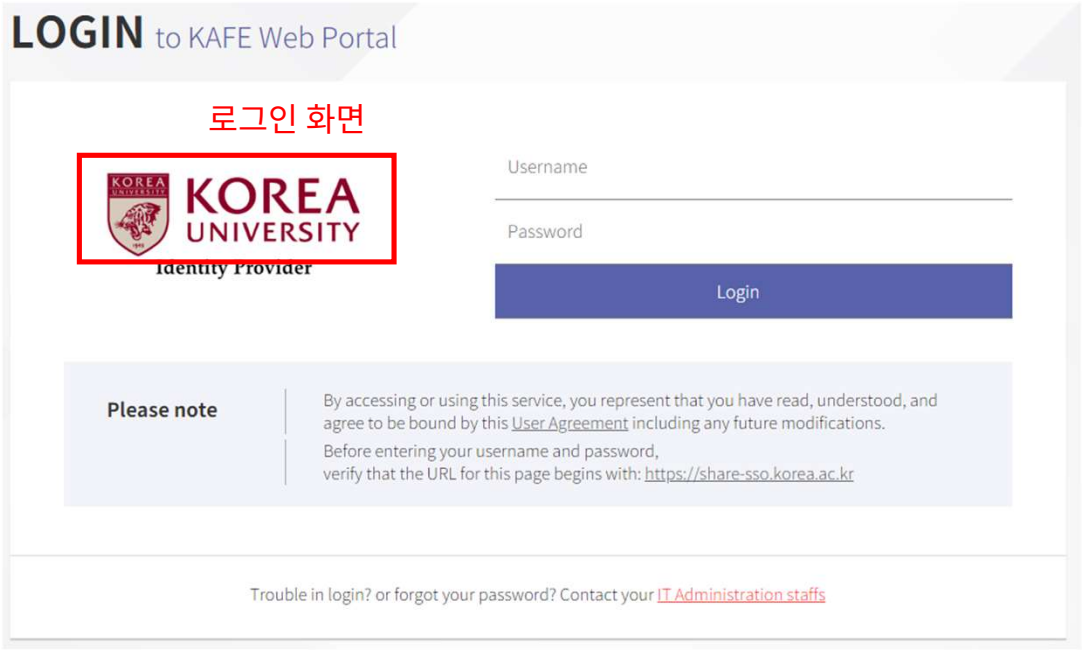
# 데이터베이스 테이블 또는 API 구성

- 필수 항목 (최소)
  - User ID, user password, 이름, 이메일, 직무정보
- User ID/password
  - (권장) email ID와 다른 ID 사용
  - (권장) email PW와 다른 PW 사용
- 이름
  - (권장) 영문으로 성과 이름을 구분
- 직무정보
  - (필수) faculty, student, staff, alum 으로만 구분
  - (권장) alum은 서비스 이용 불가

# 기타

- 기관 로고 파일
  - 탐색 서비스 화면 용: 60x20 ~ 70x25 pixel
  - 로그인 화면 용: 240x80 pixel 내외



# 설치

- Rocky linux 8.8 설치 후 IdP 서버로 SSH 접속
  - 현재 시간과 리눅스 서버 시간이 일치하는지 확인

- root 권한으로 변경

```
$ su
```

- KAFE IdP 설치 소프트웨어 다운로드
  - https://git.kafe.or.kr/kafe-software/simplesamlphp-rocky-8-9-installer.git
    - 권한 확보 필요
    - 권한 부재 시 support@kafe.or.kr에 zip 파일 제공 요청

```
$ cd ~
$ git clone https://git.kafe.or.kr/kafe-software/simplesamlphp-rocky-8-9-installer.git
```

  - ‘SSL certificate problem’ 오류 발생 시

```
$ git config --global http.sslVerify false
$ $ git clone https://git.kafe.or.kr/kafe-software/simplesamlphp-rocky-8-9-installer.git
```

  - 다운로드 성공 시 아래와 같음

```
$ ls
simplesamlphp-rocky-8.9-installer
```

# 환경 설정 - IdP 설치 소프트웨어

- Simplesamlphp-rocky-8-9-installer 디렉토리 이동 후 .env 수정

```
$ ls
simplesamlphp-rocky-8.9-installer

$ cd simplesamlphp-rocky-8.9-installer
$ nano .env
```

- .env 환경 변수 설정

```
# Local DB configuration (IdP 서버에서 관리하는 local DB, 설치 후 환경 변경 불가)
DB_USER                 # KAFE_DB_NAME DB에 접근하기 위한 사용자 ID
DB_PASSWORD             # KAFE_DB_NAME DB에 접근하기 위한 사용자 PW
KAFE_DB_NAME
USER_TEST_DB            # 테스트용 사용자 DB 명 (기관 계정 DB 연동 전 사용)

INSTALLER_PATH          # 기본값 유지

# SAML SW 설치 경로
INSTALL_PATH            # 기본값 유지
SSP_VERSION             # 2.1.4


PHP_VERSION             # 기본값 유지

MODULES_REPO            # 기본값 유지
```

# 환경 설정 - IdP 설치 소프트웨어 (계속)

- .env 환경 변수 설정 (계속)

```
# SAML SW 환경 설정
POSTFIX                     # 기본값 유지
ADMIN_PASSWORD              # SAML SW 관리자 비밀번호
ADMIN_NAME                  # SAML 소프트웨어/서버 관리자 이름, 기본값 유지
ADMIN_EMAIL                 # 서버 관리자 이메일


OTP_DB_NAME                 # 기본값 이용

# TLS 인증서 생성
DOMAIN                      # IdP 서버의 도메인 주소
SSL_CITYNAME                # 기관이 위치한 도시 이름(영문)
SSL_COMPANYNAME             # 기관이름(영문)
SSL_DEPTNAME                # 서버 관리 부서(영문)


MEMBER_ORG_KR               # 기관명(한글)
MEMBER_ORG                  # 기관명(영문, fullname)
MEMBER_ORGDISPLAY           # 기관명(영문, fullname)
MEMBER_ORGURL               # 기관 대표 홈페이지 주소

MEMBER_ORGIMG               # 기관 로고(큰 그림) 파일 명
MEMBER_ORGIMG_DS            # 기관 로고(작은 그림) 파일 명
```

# 환경 설정 - IdP 설치 소프트웨어 (계속)

- .env 환경 변수 설정 (계속)

```
# 메타데이터 환경 변수 (중요)
META_SCOPE                  # 기관 대표 도메인 명 예, kisti.re.kr, snu.ac.kr, korea.ac.kr 등
ATTRIBUTE_SCHACHOMEORG      # META_SCOPE 값과 동일하게 설정

# 이하 기본 값으로 유지
```

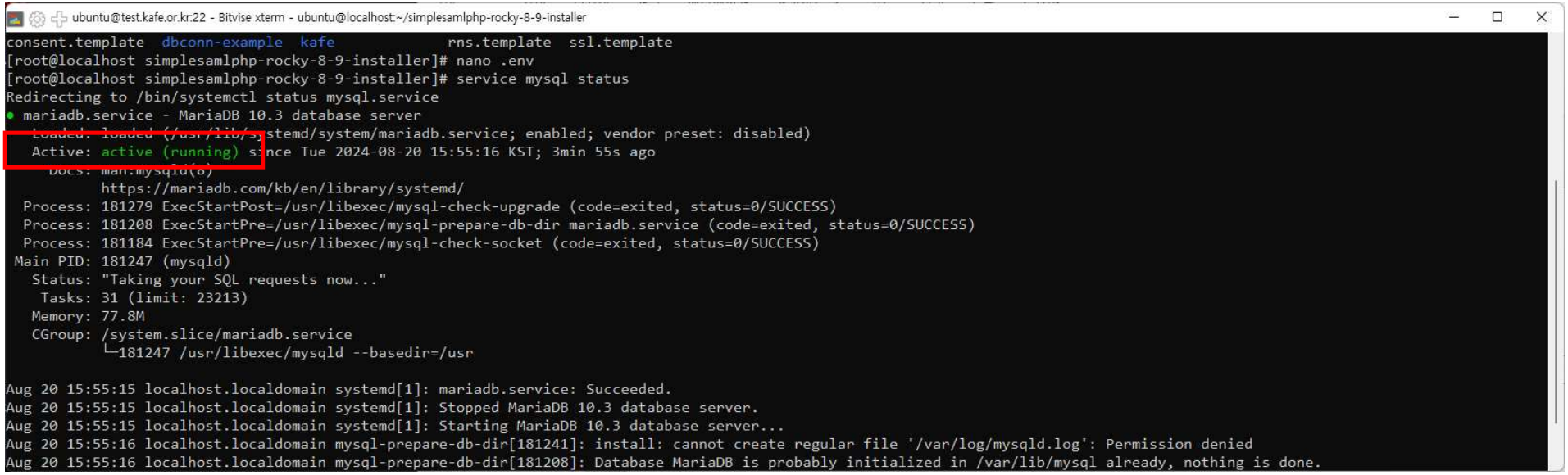
- 설치 스크립트 실행

```
$ pwd
/root/simplesamlphp-rocky-8-9-installer

# MEMBER_ORGIMG와 MEMBER_ORGIMG_DS에 설정한 로고파일 2개를 simplesamlphp-rocky-8-9-installer/kafe/theme/images 디렉토리에 복사한 후 아래 명령을 실행

$ bash install.sh
```

# 설치 검증

- MariaDB, HTTPD, PHP

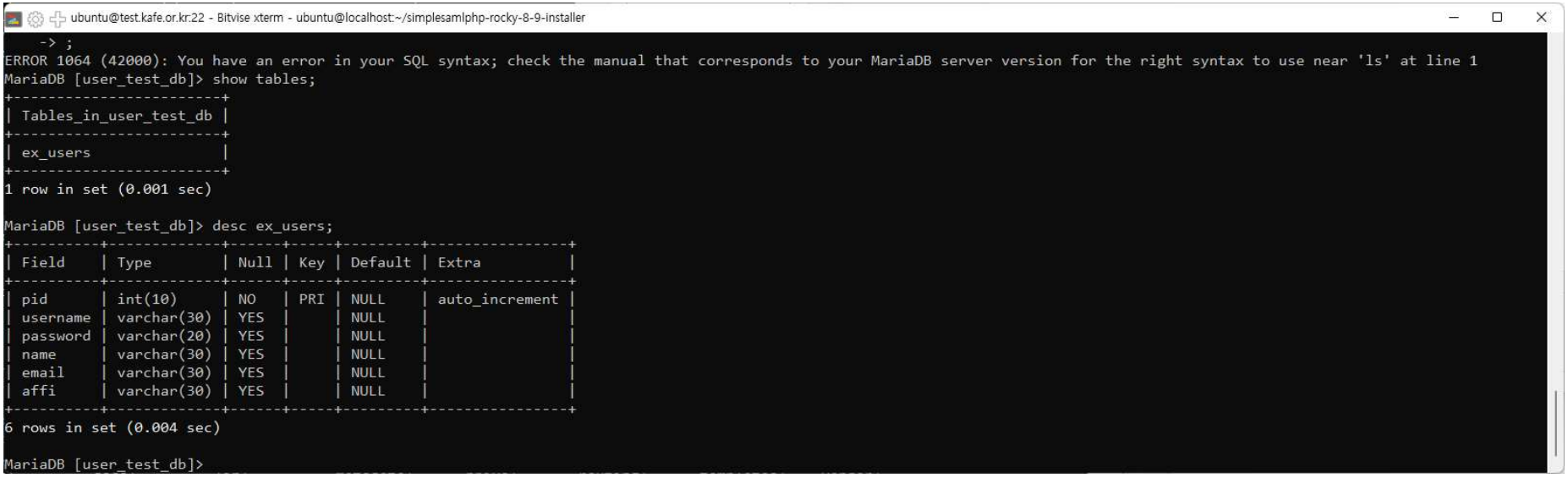
```
$ service mysqld status
$ service php-fpm status
$ service httpd status
$ service chronyd status
```



# DB table 확인

- DB 로그인

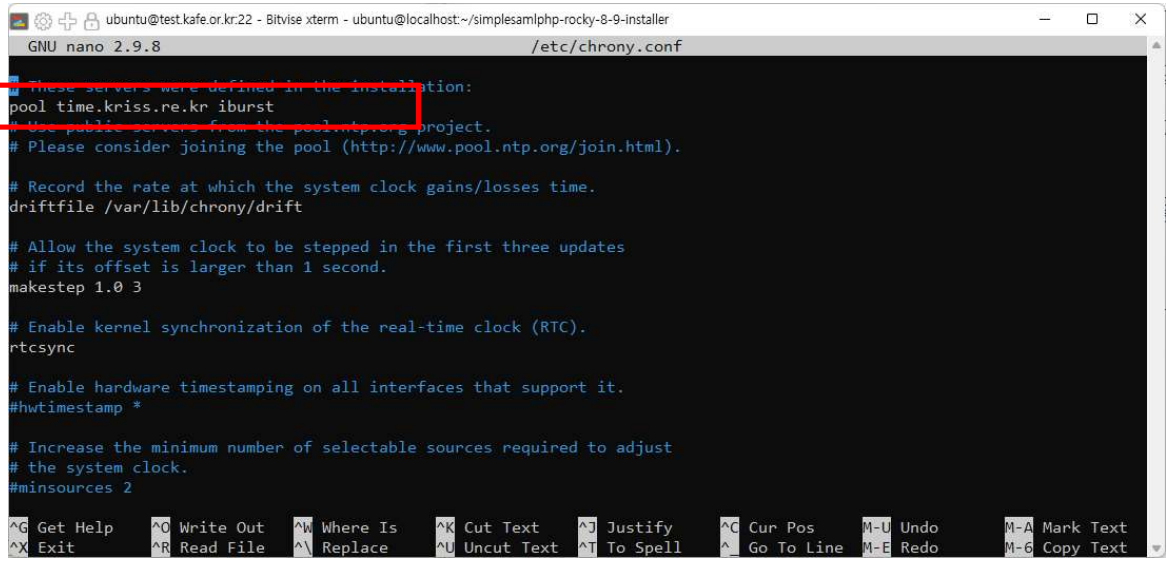
```
# .env 파일에 설정한 DB_USER 값과 DB_PASSWORD 값을 입력
$ mysql –u kafe –p
Password:

$ show databases;
$ use user_test_db;
$ show tables;
$ desc ex_users;
```



# NTP 서버 변경

- Chronyd 사용 함

```
$ nano /etc/chrony.conf

# 아래 라인을 찾아 NTP 서버를 변경
pool time.kriss.re.kr iburst

$ systemctl restart chronyd
```

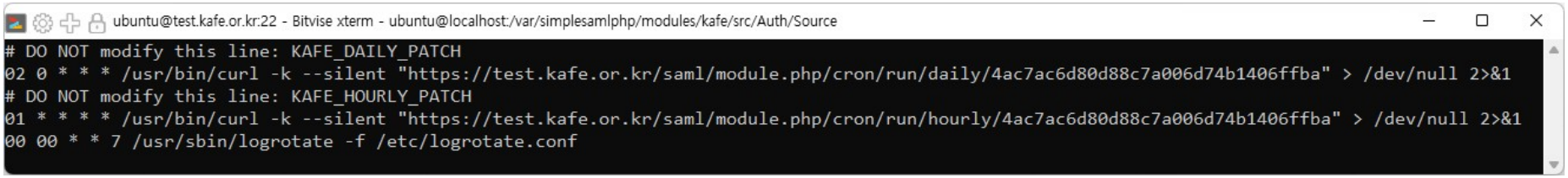


# Cron 확인

- Cron을 이용해 연합 메타데이터를 주기적으로 다운로드 함

```
$ crontab -e
```

  - IdP의 도메인이 test.kafe.or.kr인 경우




- 연합 메타데이터의 다운로드 주소 확인

```
$ nano /var/simplesamlphp/config/module_metarefresh.php
```

KAFE

# HTTP 서비스 확인

- .env 파일에 DOMAIN 변수 값으로 설정한 주소로 접근

# 공인 TLS 인증서 설치

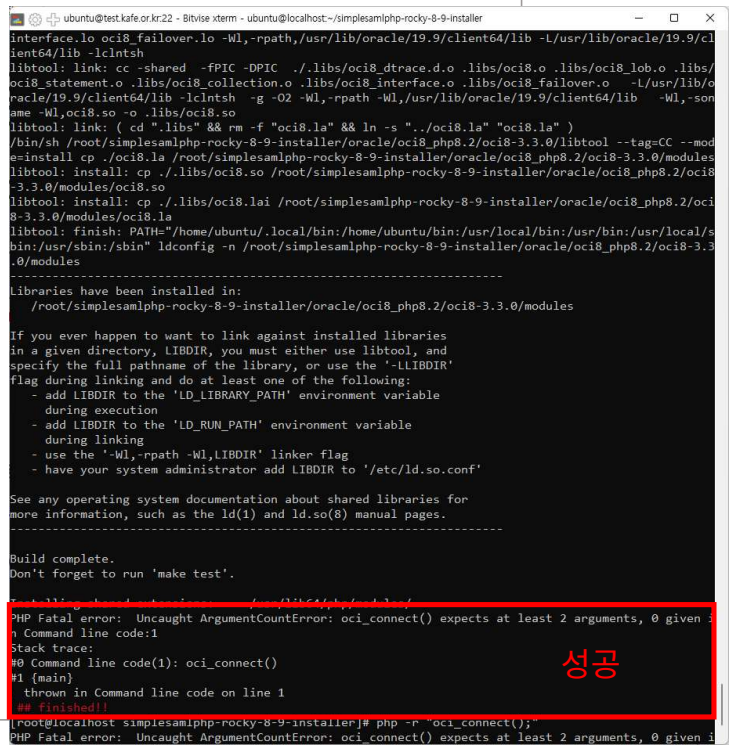
- 환경 설정

```
$ nano /etc/httpd/conf.d/ssl.conf

SSLCertificateFile /etc/pki/tls/test.kafe.or.kr/ca.crt              # 기관 wildcard 인증서로 cert로 변경
SSLCertificateKeyFile /etc/pki/tls/test.kafe.or.kr/ca.key           # 기관 wildcard 인증서의 key로 변경
#SSLCertificateChainFile /etc/pki/tls/certs/server-chain.crt        # 필요 시 변경
#SSLCACertificateFile /etc/pki/tls/certs/ca-bundle.crt              # 필요 시 변경

# 설정 후
$ systemctl restart httpd
```

# 기관 계정 DB 연동 검증 – DBMS 드라이버 설치

- 설치된 초기 IdP 소프트웨어는 IdP 서버 내 Local DB에 연동된 상태 → 기관 계정 DB로 연동 변경 필요
- IdP 서버에 기관 계정 DB의 DBMS Client driver 설치(예, Oracle OCI client)

```
# simplesamlphp-rocky-8-9-installer의 dbconn_setup.sh의 oracle 부분을 찾아 수동 설치(스크립트 설치 시 repository 접근 권한 필요)
# oci8은 3.3.0 을 설치(3.3.0 버전이 php 8.2와 호환)
# 아래 예시는 기관 계정 DB의 DBMS가 Oracle일 경우에 한 함 → 기관 DBMS에 맞는 driver 설치

$ dnf install libnsl libssh2-devel -y
$ dnf install systemtap-sdt-devel –y

# oracle instant  client basic과 devel을 oracle 홈페이지에서 다운로드
$ rpm -ivh oracle-instantclient19.9-basic-19.9.0.0.0-1.x86_64.rpm
$ rpm -ivh oracle-instantclient19.9-devel-19.9.0.0.0-1.x86_64.rpm

# oci 8을 설치 (https://pecl.php.net/package/oci8에서 oci8-3.3.0.tgz을 다운로드)
$ tar xvzf oci8-3.3.0.tgz
$ cd ./oci8-3.3.0
$ phpize
$ export PHP_DTRACE=yes
$ ./configure --with-oci8=shared,instantclient,/usr/lib/oracle/19.9/client64/lib --enable-shared
$ ln -s /usr/include/oracle/19.9/client64/ /usr/lib/oracle/19.9/client64/lib/include
$ make all install
$ echo 'extension=oci8.so' > /etc/php.d/oci8.ini
$ systemctl restart php-fpm
$ systemctl restart httpd
$ php -r "oci_connect();"
```



# 기관 계정 DB 연동 검증 – 검증용 Script 작성

- 테스트용 php script를 작성

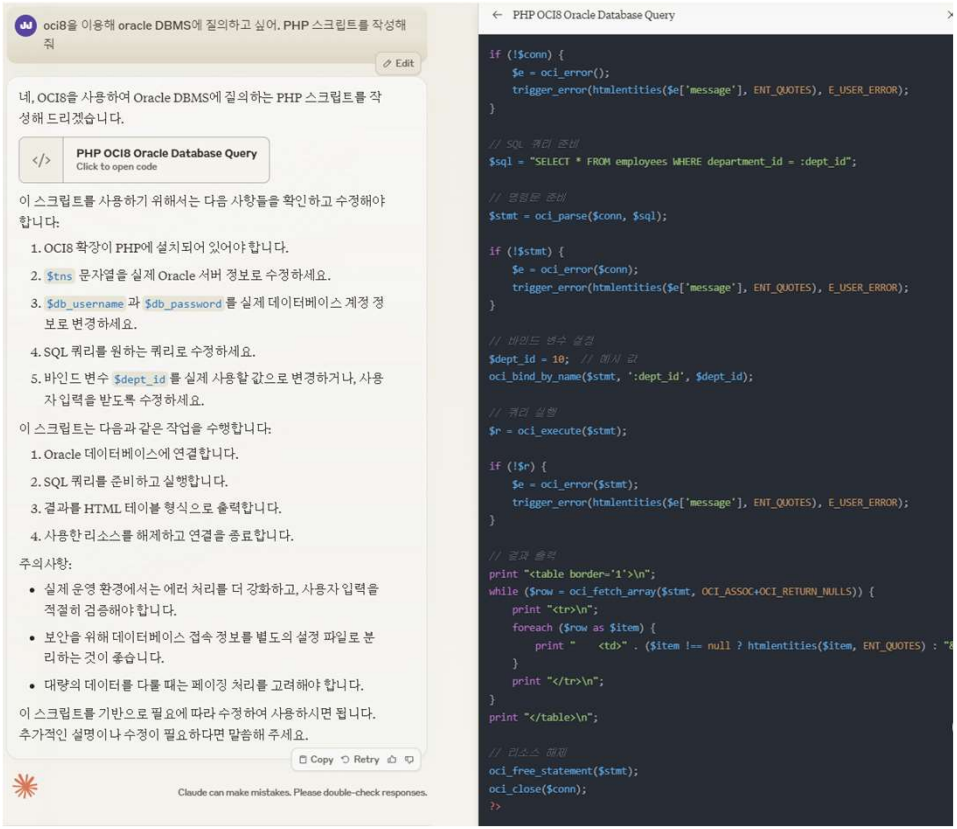
```
# simplesamlphp-rocky-8-9-installer/dbconn-example/oci_connect.php를 참조
$ nano oci_connect.php
<?php
  $db_username='DB_USER_NAME';
  $db_userpass ='DB_USER_PASS';
  $username='USER_NAME';
  $password='USER_PASS';

  # tns(Transparent Network Substrate) string
  $db = "(DESCRIPTION=(ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = DB_IP_ADDR)(PORT = DB_PORT)))(CONNECT_DATA=(SID=SID_NAME)))";

  $conn = oci_connect($db_username, $db_userpass, $db);
  if(!$conn){ $e = oci_error(); var_dump($e); }

  # 비밀번호 암호화 알고리즘을 고려해 질의해야 함
  $stid = oci_parse($conn, "SELECT * FROM TABLENAME_ORG WHERE USERNAME_ORG='$username' and PASSWORD_ORG='$password'");
  oci_execute($stid);

  if(!$row = oci_fetch_assoc($stid)){ echo "error"; exit; }

  var_dump($row);
  oci_free_statement($stid);
  oci_close($conn);
?>
```

# 기관 계정 DB 연동 검증 –Script 동작 검증

- IdP 서버와 기관 계정 DB 간 연동 검증

```
$ php oci_connect.php

# 사용자 정보가 정상적으로 출력되면 OK
```

<참고> claude.ai 또는 chat.openai.com를 이용해 원시 script 작성 가능
예, ‘oci8을 이용해 oracle DBMS에 질의하고 싶어. PHP 스크립트를 작성해줘’
예, ‘API 호출 경로가 https://kafe.or.kr/authencation/users?id=username&pw=123 임. PHP 8.2에서 동작하는 스크립트를 작성해줘”



# IdP와 기관 계정 DB 연동 (계속)

- 기관 계정 DB 연동 및 로그인 시도 횟수 제한

```
$ nano /var/simplesamlphp/modules/kafe/src/Auth/Source/CoreAuth.php

# 데이터 베이스 연결설정과 질의 등(반환 값을 얻어오는 부분까지)은 oci_connect.php를 참조해 설정
# DB 연결 설정과 관련된 오류는 'throw new Exception()'을 이용해 예외 처리 함
…,
$row = oci_fetch_assoc($stid));

if(!$row) $tmp_userinfo = false;
else {
        $tmp_userinfo = [
          'name'  => $row['name'],
          'email' => $row['email']
        ];
}

# 사용자 잠금 확인(blockCheck()) 및 로그인 시도횟수 초과 사용자 잠금(wrongUserpass())
$kafeUtil = new kafe\Util();
$this->blockCheck($kafeUtil, $username, $tmp_userinfo);
If (!$row) {
        /* User not found */
        Logger::warning( ' MyAuth: Could not find user '  . Var_export($username, TRUE) .  ' . ' );
        $this->wrongUserpass($kafeUtil, $username, []);
}
/* 생략 If (!$this->checkPassword($row[ ' password ' ], $password)) { …,} */

// Login success
$kafeUtil->resetWrongCount($username);
```

$username: 사용자 ID (사용자 이름 아님)

계정 DB에서 반환되는 배열을 $row로 가정
$row에 사용자 속성값이 저장됨

$row['affi']:  사용자의 직무정보
$row['name']: 사용자의 성명
$row['email']: 사용자의 전자우편 주소

# IdP와 기관 계정 DB 연동 (계속)

- 사용자 속성값 확보 및 생성

```
# 직무 정보가 affiliationMap에 존재하면, 반환 받은 값에 member를 추가해 eAffiliation에 저장
# 예, ‘faculty’를 반환 받으면 [‘faculty’, ‘member’]로 변경

# eduPersonAffiliation 생성
    if (array_key_exists($row[ ' affi ' ], $this->affiliationMap)) {
        $this->eAffiliation = $this->affiliationMap[$row[ ' affi ' ]];
    } else {
        $this->eAffiliation = [ ' affiliate ' ];
    }

# 직무정보가 alum인 사용자 차단
    if (in_array('alum', $this->eAffiliation)) {
        Logger::warning('KafeAuth: the user ' . var_export($username, TRUE) . ' has no privilege.');
        throw new Error('NOPRIVILEGE');
    }

    $config = Configuration::getOptionalConfig('module_kafe.php')->toArray();

# sn과 givenName 생성: 성명을 ‘성 ' 과 ‘이름 ' 으로 분리
    $name = $this->separateName($row['name']);
    $sn = [$name['sn’]];   # 성
    $givenname = [$name[‘gn’]];  #이름

# eduPersonScopedAffiliation 생성: scope 내 직무정보를 표기 예, faculty@kafe.or.kr
# 구축한 IdP의 scope이 “kafe.or.kr”인 경우
    $epsAffiliation = [$this->eAffiliation['0'] . "@kafe.or.kr"];
```

$username: 사용자 ID (사용자 이름 아님)

계정 DB에서 반환되는 배열을 $row로 가정
$row에 사용자 속성값이 저장됨

$row[‘affi’]:  사용자의 직무정보
$row[‘name’]: 사용자의 성명
$row[‘email’]: 사용자의 전자우편 주소

```
$this->affiliationMap = array(
        …,
        'student' => array('student', 'member'),
        'staff' => array('staff', 'member'),
        'faculty' => array('faculty', 'member'),
        'alum' => array('alum'),
    );
```

# IdP와 기관 계정 DB 연동 (계속)

- 사용자 속성값 확보 및 생성

```
# eduPersonPrincipalName 생성
# 구축한 IdP의 scope이 “kafe.or.kr”이고
# sha1으로 $username을 encryption 한 경우
 $eppName = [sha1($username) . "@kafe.or.kr”];

# 모든 속성값은 배열([]) 형태로 저장
 $attributes = [
        ' uid '  => [$username],
        ' displayName '  => [$row["name"]],  #성명과 동일
        ' email '  => $row[ " email " ],
        ' sn '  => $sn,
        ' givenName '  => $givenname,
        ' eduPersonAffiliation '  => $this->eAffiliation,
        ' eduPersonPrincipalName '  => $eppName,
        ' eduPersonScopedAffiliation '  => $epsAffiliation,
        ' mail '  => $row[ " email " ],                          #email과 mail의 값은 동일
        ' organizationName '  => ["Korean Access Federation"],   # 기관 이름이 “Korean Access Federation”인 경우
        ' schacHomeOrganization' => ["kafe.or.kr"],              # IdP의 scope이 “kafe.or.kr”인 경우

…,
```

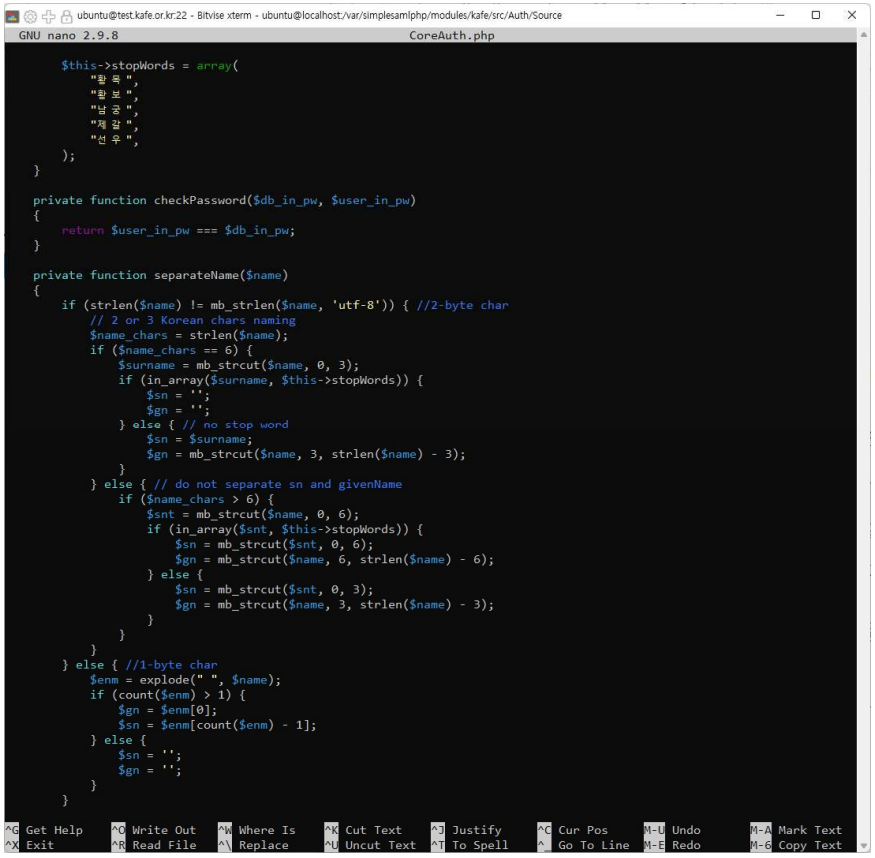
$username: 사용자 ID (사용자 이름 아님)

계정 DB에서 반환되는 배열을 $row로 가정
$row에 사용자 속성값이 저장됨

$row[‘affi’]:  사용자의 직무정보
$row[‘name’]: 사용자의 성명
$row[‘email’]: 사용자의 전자우편 주소

# 성명의 분리

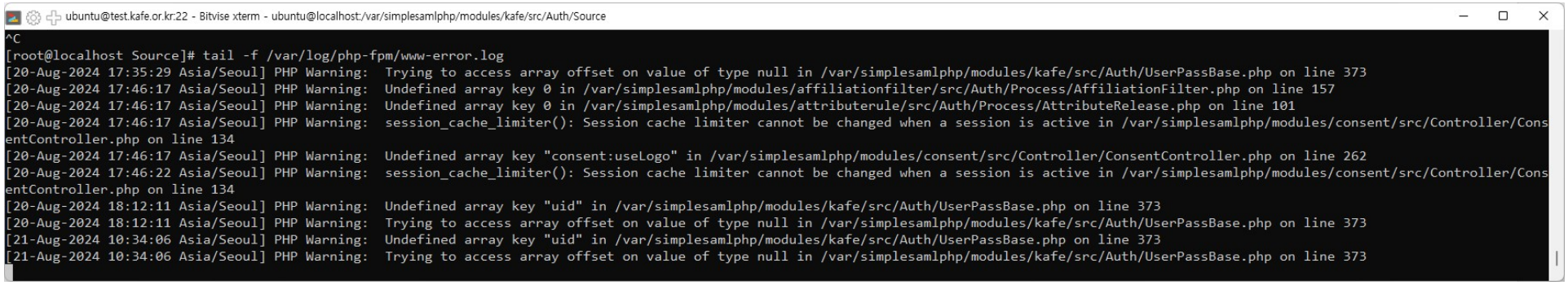
- 기관 계정 DB에서 사용자의 '성'과 '이름'을 구분하지 않을 경우
  - IdP 소프트웨어가 자동 구분 → 성과 이름의 구분이 실제와 다를 수 있음



# 디버깅 정보

- Linux log files

```
# /var/log/php-fpm/www-error.log
# /var/log/messages
# /var/log/httpd/ssl_access_log
# /var/log/httpd/ssl_error_log
```



KAFE

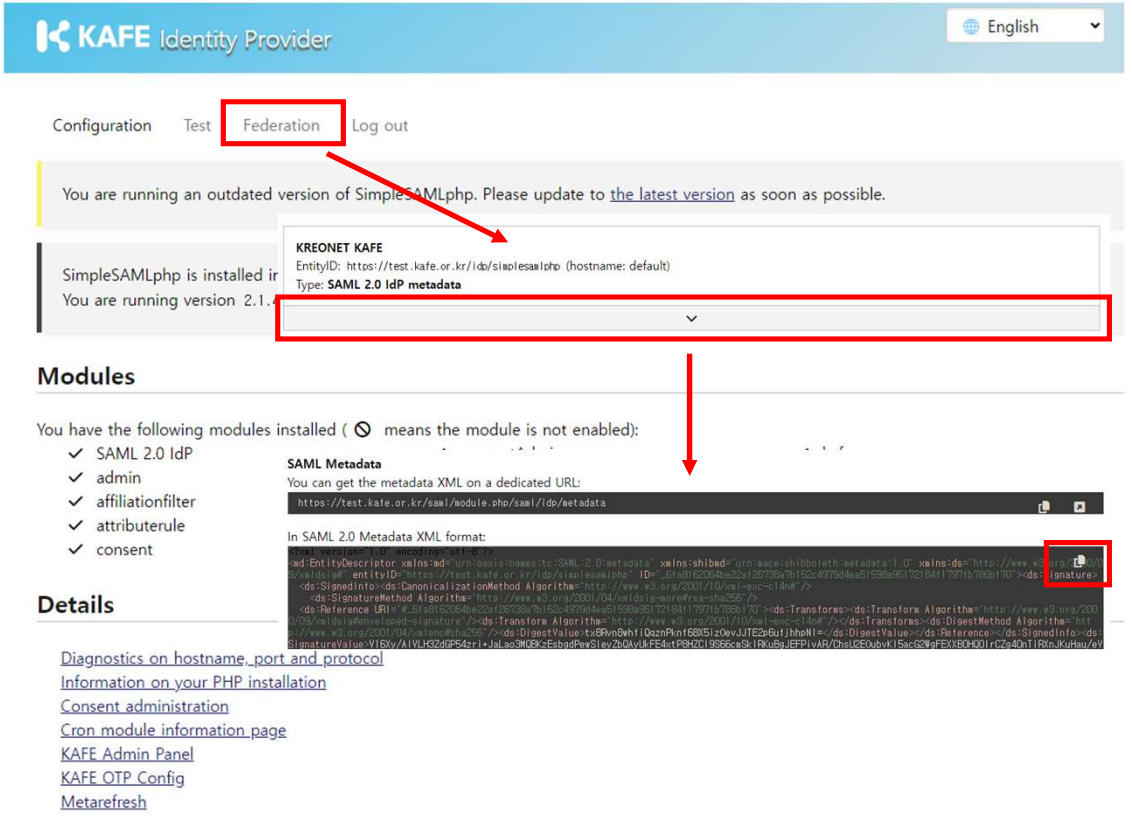
# <참고> 탐색 서비스용 로고 파일의 위치

- 서버 도메인이 test.kafe.or.kr이고 로고 파일(작은 그림)이 my_inst_logo.png 이면
- 로그파일의 위치 URL은 https://test.kafe.or.kr/my_inst_logo.png 임
  → 탐색서비스 로고 표시에 활용되므로 KAFE에 위치 URL을 알려야 함

# 3교시: IdP의 검증/관리와 KAFE 가입

# 구축된 IdP의 검증

- IdP의 메타데이터 확보
  1. https://[test.kafe.or.kr]/saml/module.php/admin
     - 암호는 .env 파일의 ADMIN_PASSWORD 값
  2. Federation 탭 클릭
  3. SAML 2.0 IdP metadata 클릭
  4. In SAML 2.0 Metadata XML format의 메타데이터를 복사



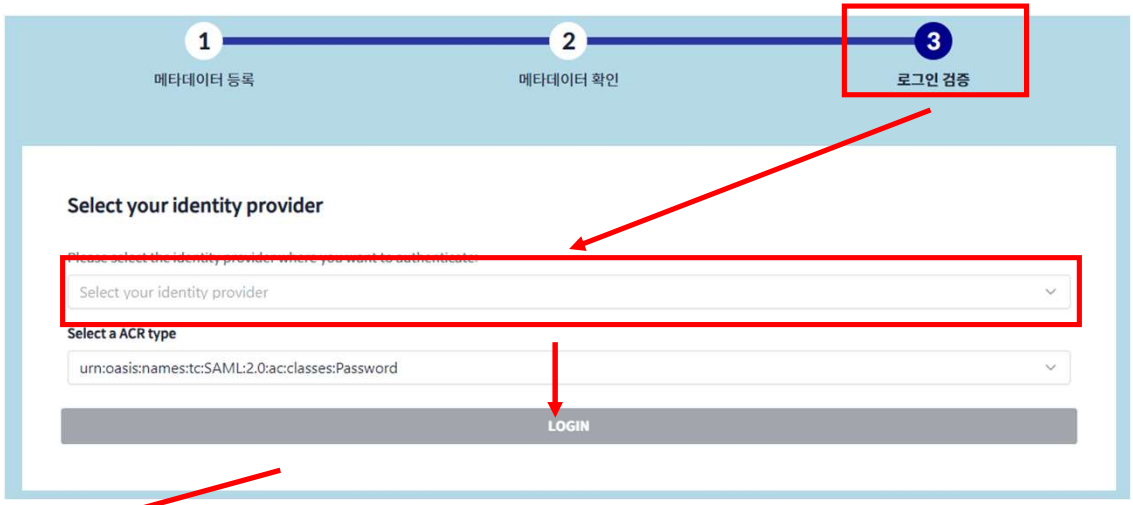
# 구축된 IdP의 검증 (계속)

- KAFE debugger에 IdP 메타데이터 등록
  - https://debug.kafe.or.kr
  - SAML 선택
  - Input raw xml에 IdP 메타데이터를 붙여넣기

<참고> debug.kafe.or.kr은 IdP를 검증하기 위한 검증용 서비스제공자 임

# 구축된 IdP의 검증 (계속)

- 로그인 검증으로 이동
  - 구축된 IdP를 선택하고 로그인
  - 사용자 정보가 전달되는지 확인




AuthnContextClassRef

| Requested | Returned |
|---|---|
| urn:oasis:names:tc:SAML:2.0:ac:classes:Password | urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport |

NameID

| Value | Format |
|---|---|
| _048f1f12545ed132836baa8bc85816449b44d404e1 | urn:oasis:names:tc:SAML:2.0:nameid-format:transient |

SAML Attributes

| Attribute | Value |
|---|---|
| displayName | 홍길동 |
| eduPersonAffiliation | student |
| eduPersonEntitlement | urn:mace:dir:entitlement:common-lib-terms |
| eduPersonPrincipalName | 2439e0457579ab4fd962cbd80b9206aca794cc38@kafe.or.kr |
| eduPersonScopedAffiliation | student@kafe.or.kr |
| eduPersonTargetedID | c3bbfc19e9cf98ff6548bc498a6e1f5e37f69a07 |
| mail | coreen@example.org |
| o | KREONET KAFE |
| schacHomeOrganization | kafe.or.kr |
| uid | student1 |

2명 이상의 사용자를 대상으로 검증: eduPersonTargetedID 값이 서로 달라야 함

# KAFE 가입 등 회원기관 등록 절차

1. (가입 기관) IdP 가입신청서 작성 및 제출
   - 가입 신청서는 https://www.kafe.or.kr/join에서 'ID 제공자용 프로덕션 페더레이션 참가 신청서'를 다운로드
   - IdP 가입신청서는 기관장(원장, 총장 등) 직인 필수
   - 공문(수신처: 한국과학기술정보연구원) 또는 support@kafe.or.kr로 제출
     - 구축된 IdP의 메타데이터와 탐색서비스용 로고 파일의 위치를 함께 제출

2. (KISTI) KAFE Metadata Registry 시스템에 메타데이터 등록
   - 등록 후 연합 메타데이터가 배포되는데 하루 이상 소요

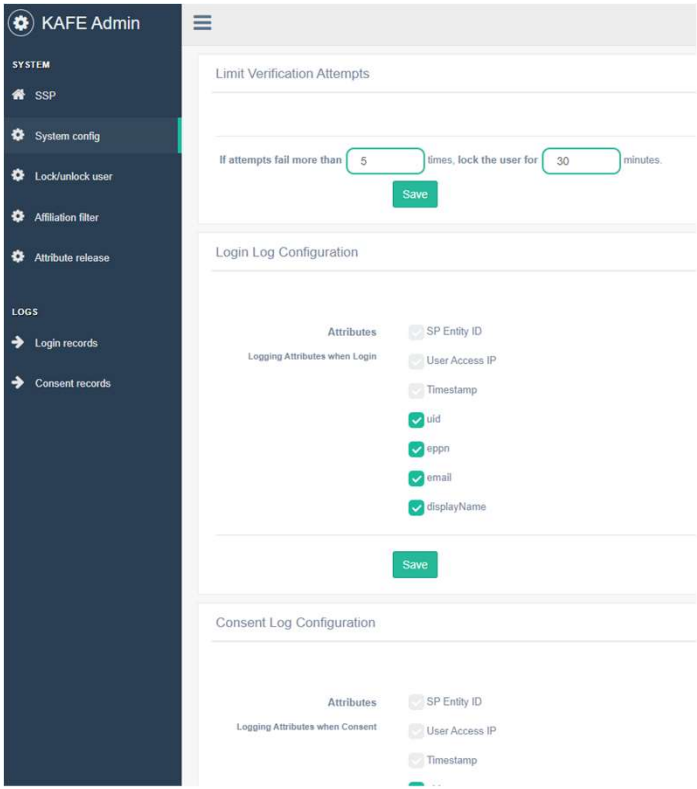
3. (가입 기관) 프로덕션 중인 서비스제공자를 대상으로 로그인 검증 실시
   <참고> 상용 서비스 이용 시 기관 구독 및 별도 연결 설정 필요

# IdP의 관리

- KAFE Admin Panel과 KAFE OTP Config를 이용해 IdP 관리
  - https://[test.kafe.or.kr]/saml/module.php/admin

# KAFE Admin Panel

- 메뉴 설명
  - System config
    - 로그인 시도 횟수 및 로그 정보 설정
  - Lock/unlock user
    - 사용자 ID를 기준으로 차단 및 차단해제
  - Affiliation filter
    - 특정 서비스에 대한 접근을 직무정보를 기반으로 제어
  - Attribute release
    - 특정 서비스에 제공할 사용자 속성을 정의
  - Login records
    - 로그인 기록 확인
  - Consent records
    - 사용자 동의 기록을 확인



# KAFE Admin Panel - Affiliation filter

- 특정 서비스제공자에 접속 허용할 사용자의 직무정보를 설정하는데 이용
- 이용 방법
  - 메타데이터에서 서비스제공자의 고유식별자 확인
    - /var/simplesamlphp/config/module_metarefresh.php에서 구축된 IdP가 사용 중인 연합 메타데이터를 확인 가능

```
# 예
'src' => 'https://fedinfo.kreonet.net/signedmetadata/federation/KAFE-testfed/metadata.xml',
'src' => 'https://fedinfo.kreonet.net/signedmetadata/federation/KAFE-eduGAIN/eduGAIN-SP.xml'
```

  - 웹 브라우저에서 URL을 입력한 후, 설정하고자 하는 서비스제공자를 검색(예, thomson)
  - 해당 서비스제공자의 개체식별자(entityID)를 확인

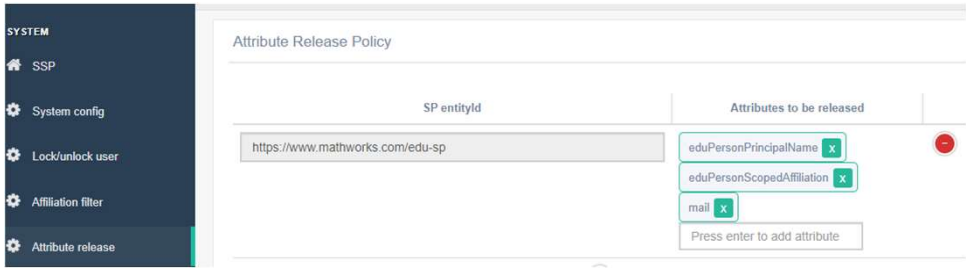
```
<!--
https://onepass.thomsonreuters.com/entity
-->
<md:EntityDescriptor entityID="https://onepass.thomsonreuters.com/entity">
  <md:Extensions>
    <mdrpi:RegistrationInfo registrationAuthority="http://kafe.kreonet.net" registrationInstant="2022-03-08T05:05:00Z"/>
    <mdattr:EntityAttributes>
      <saml:Attribute Name="http://kafe.kreonet.net/jurisdiction" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml:AttributeValue>US</saml:AttributeValue>
      </saml:Attribute>
    </mdattr:EntityAttributes>
  </md:Extensions>
```

  - 서비스제공자의 entityID를 선택하고, 해당 서비스에 접근할 수 있는 사용자의 직무정보를 입력

# KAFE Admin Panel - Attribute release

- SP 메타데이터에 포함된 요구 속성과 다른 속성을 IdP가 제공하고자 하는 경우 이용
  - 예, mathworks는 eduPersonPrincipalName 또는 eduPersontargedID 중 하나의 속성을 필수 요구 하지만, metadata에는 위 속성이 필수 속성으로 지정되어 있지 않음 → IdP는 필수 속성 (Required attributes)만 제공 함

| | |
|---|---|
| SUPPORTED PROTOCOLS: | • SAML 1.1<br>• SAML 2.0 |
| REQUIRED ATTRIBUTES: | • urn:oid:1.3.6.1.4.1.5923.1.1.1.9 (eduPersonScopedAffiliation)<br>• urn:oid:0.9.2342.19200300.100.1.3 (mail) |
| OPTIONAL ATTRIBUTES: | • urn:oid:1.3.6.1.4.1.5923.1.1.1.6 (eduPersonPrincipalName)<br>• urn:oid:1.3.6.1.4.1.5923.1.1.1.10 (eduPersonTargetedID)<br>• urn:oid:2.5.4.42 (givenName)<br>• urn:oid:2.5.4.4 (sn) |



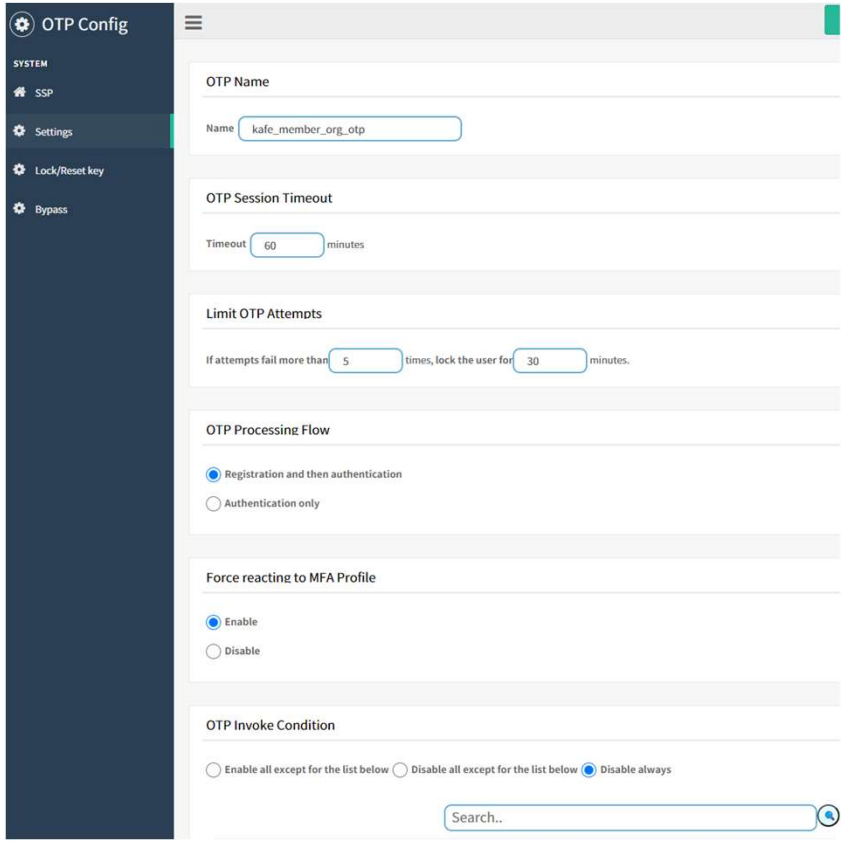
# KAFE OTP config

- 내장 TOTP 또는 외장(상용) MFA 이용 가능
  - 상용 MFA 이용 권장
    - RESTful API를 지원해야 함
  - 내장 TOTP는 Google OTP와 호환

- 내장 TOTP 이용 설정

```
$ nano /var/simplesamlphp/config/config.php

# 'authproc.idp'를 검색
# 아랫 라인의 주석(//)을 해제
//89 => 'kafeotp:OTP',
```

  - OTP Config 환경 설정
    - OTP Processing flow: 기본값 유지
    - Force reacting to MFA profile: 기본값 유지
    - OTP invoke condition
      - 리스트에 명시된 SP를 제외하고 모든 SP에 대해서 활성화
      - 리스트에 명시된 SP만 활성화
      - 모든 SP에 대해서 비활성화

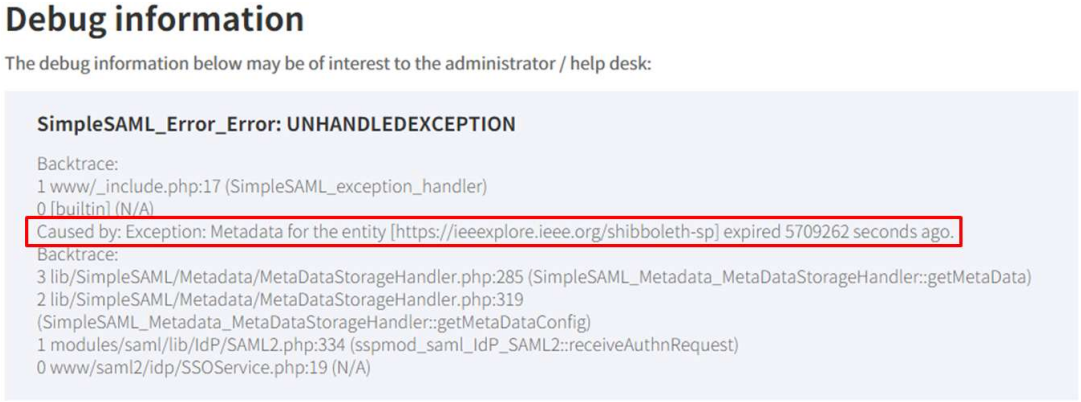


# 오류 관리

- 데몬 서비스 상태
  - httpd, php-fpm, mysqld, chronyd가 항상 active (running) 상태로 유지되어야 함

```
$ systemctl status httpd
$ systemctl status php-fpm
$ systemctl status mysqld
$ systemctl status chronyd
```

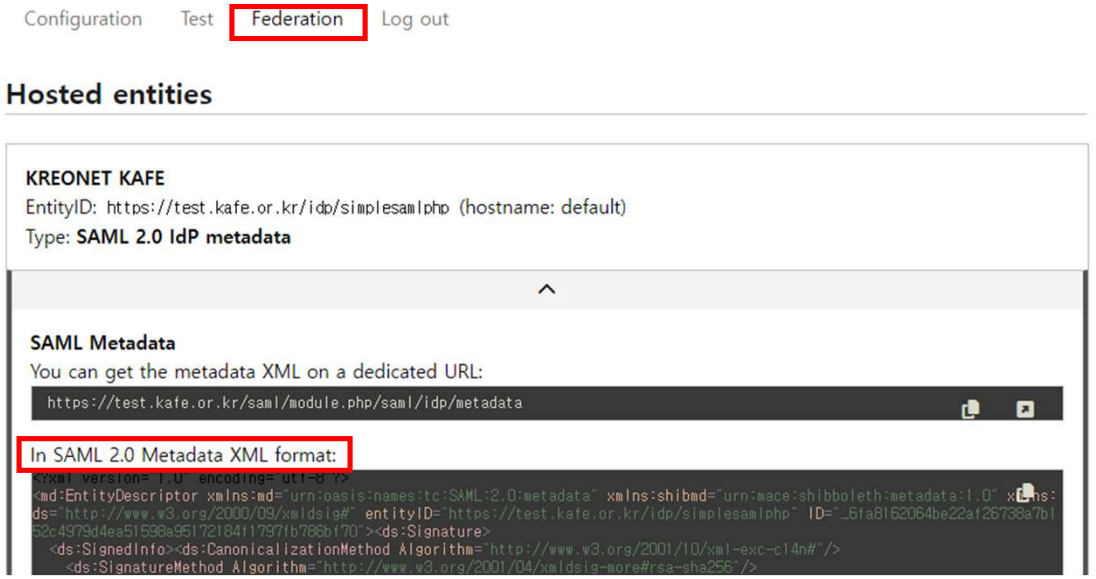
- ‘Metadata for the entity expired’ 오류
  - NTP 동기화 또는 metadata 업데이트 불량
  - Admin 화면에서 Metarefresh 클릭


- ‘Metadata not found’ 오류 (무시)
  - KAFE에 가입되어 있지 않은 서비스제공자 접근 시 발생
  - KAFE에서 다운로드를 거부한 국외 서비스제공자 접근 시 발생

# 정보 제공

- 상용서비스 연계 시 서비스제공자에 전달해야 하는 필수 정보
  - entityID, scope, affiliation 정보
    - entityID: 개체식별자
    - Scope: 기관 대표 도메인 (.env 파일 참조)
    - Affiliation: 서비스에 접근을 허용하고자 하는 사용자의 직무정보
  - IdP admin 화면의 Federation 메뉴 → SAML metadata 탭 → XML format에서 검색 가능
    - ‘entityid’, ‘scope’으로 검색
  - Affiliation
    - 학생: student@scope
      - e.g., student@kafe.or.kr
    - 교수: faculty@scope
      - e.g., faculty@kafe.or.kr
    - 직원: staff@scope
      - e.g., staff@kafe.or.kr



# Korean Access Federation

https://www.kafe.or.kr/
support@kafe.or.kr