OpenAM 学認設定ガイド

OSSTech(株)

更新日 2023 年 4 月 28 日
リビジョン 1

目次

1 はじめに

1.1 本書の目的

本書は OpenAM の環境を学術認証フェデレーションに参加し、利用するための設定に関する手順書です。

本書に関する記載内容について、疑問点等がある場合には、弊社サポート窓口までお問い合わせください。

1.2 略語

本書では必要に応じて以下の略語を用います。

- 「学術認証フェデレーション」を「学認」と表記します。

1.3 OpenAM の学認対応

OpenAM 14.5 より、学認との連携機能として次の機能が備わっています。

- 送信属性同意機能
- メタデータ自動更新機能
- computedID(eduPersonTargetedID) 生成機能

従来では学認との連携に Shibboleth IdP サーバーを導入し運用していたような構成に対し、OpenAM を導入して学認連携を OpenAM に任せることにより、 Shibboleth IdP サーバーを無くすことが出来ます。

1.3.1 制限事項

OpenAM では、全ての Shibboleth IdP の機能が利用できるわけではありません。 例えば、下記の機能は OpenAM には用意されておりません。これらの機能を利用する必要がある場合は、学認との連携は Shibboleth IdP で行い OpenAM と Shibboleth IdP を連携する構成をご検討ください。

- データベースを利用した機能
  - eduPersonTargetedID の DB 保存 (storedId)
  - 送信属性同意の情報を DB に保存
- 利用規約の表示 (terms of use)

- Shibboleth1.x の通信とバックチャネル

1.4 Shibboleth IdP サーバーからの移行

本書では、すでに学認に参加している Shibboleth IdP サーバーを OpenAM へ移行することを中心に説明します。

以降、OpenAM の環境は構築済みで認証やデータストアなどの設定は完了しているものとし、学認と連携するための設定について説明します。 OpenAM の構築に関しては初期設定ガイドや管理者マニュアル等を参照してください。OpenAM のバージョンは 14.5 以降であることが前提となります。

- 移行前の構成

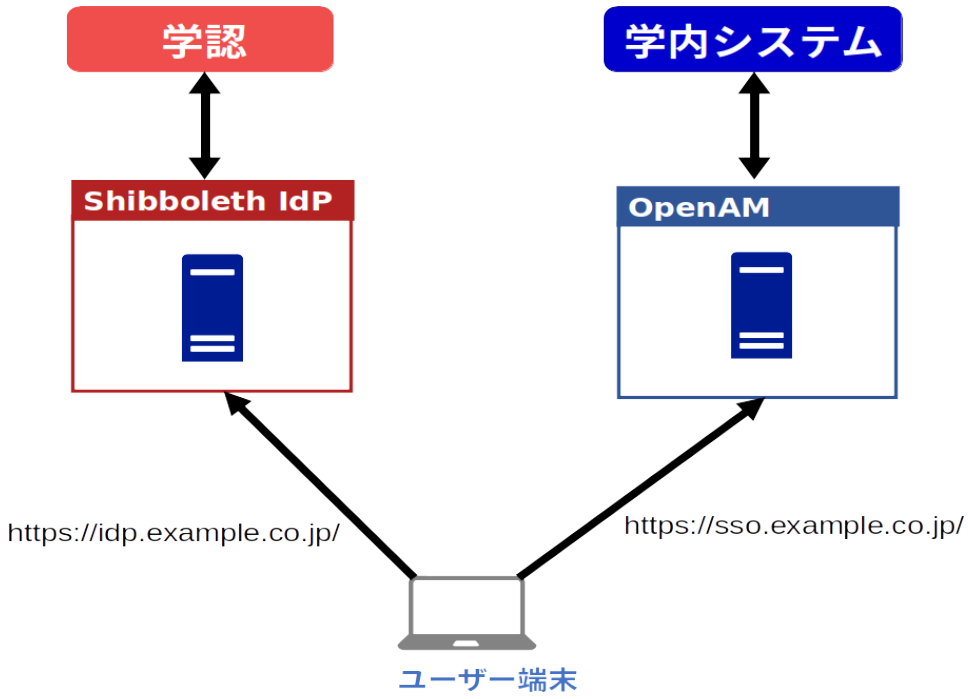

図 1　移行前の構成

移行前の構成例として、ユーザーは Shibboleth IdP を経由して学認を利用し、学内のサービスは OpenAM を経由して利用している想定とします。ここでは、Shibboleth IdP と OpenAM でそれぞれアクセスする URL が異なります。 (図では idp.exmaple.co.jp と sso.exmaple.co.jp)

- 移行後の構成

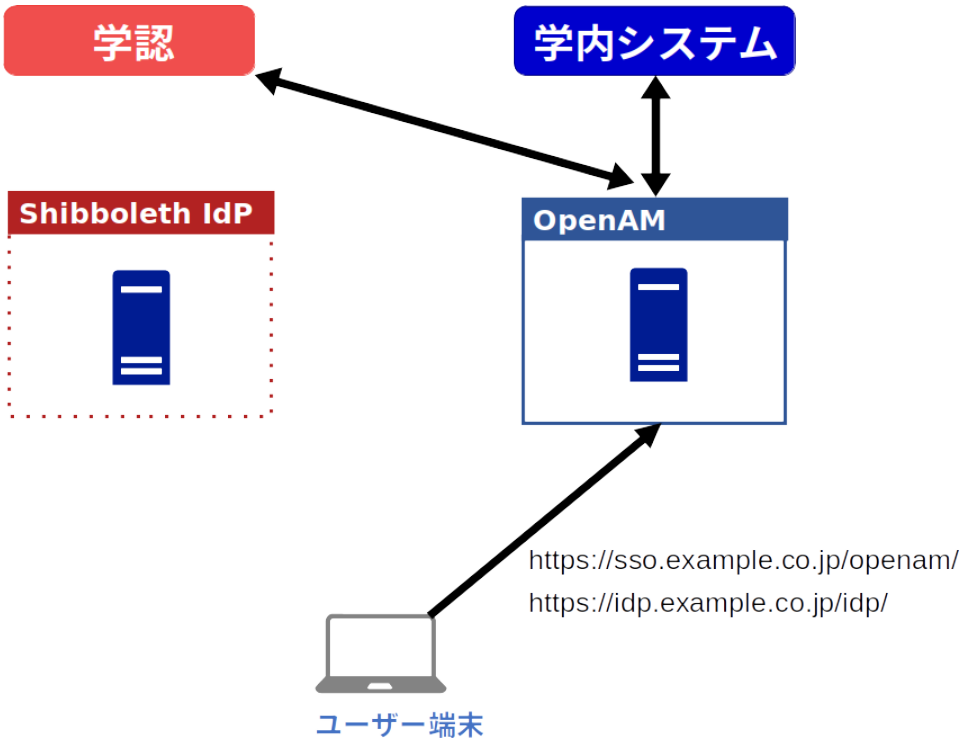

図 2　移行後の構成

移行後の構成では、ユーザーは学認も学内のサービスも OpenAM を経由して利用します。移行にあたっては、Shibboleth IdP の FQDN(図では idp.exmaple.co.jp) の名前解決を OpenAM サーバーに変更します。 つまり、Shibboleth IdP で利用していた SAML のエンドポイントの URL を OpenAM がそのまま引き継いで処理する形です。これにより、学認に登録している IdP のメタデータの変更は必要ありません。

移行の作業中の段階では、OpenAM と Shibboleth IdP の共存が可能です。Shibboleth IdP サーバーで学認を利用しつつ並行して OpenAM の学認参加の設定を行い、特定の端末 (hosts ファイルの変更) のみ OpenAM 経由で学認サービスの利用の確認が出来ます。

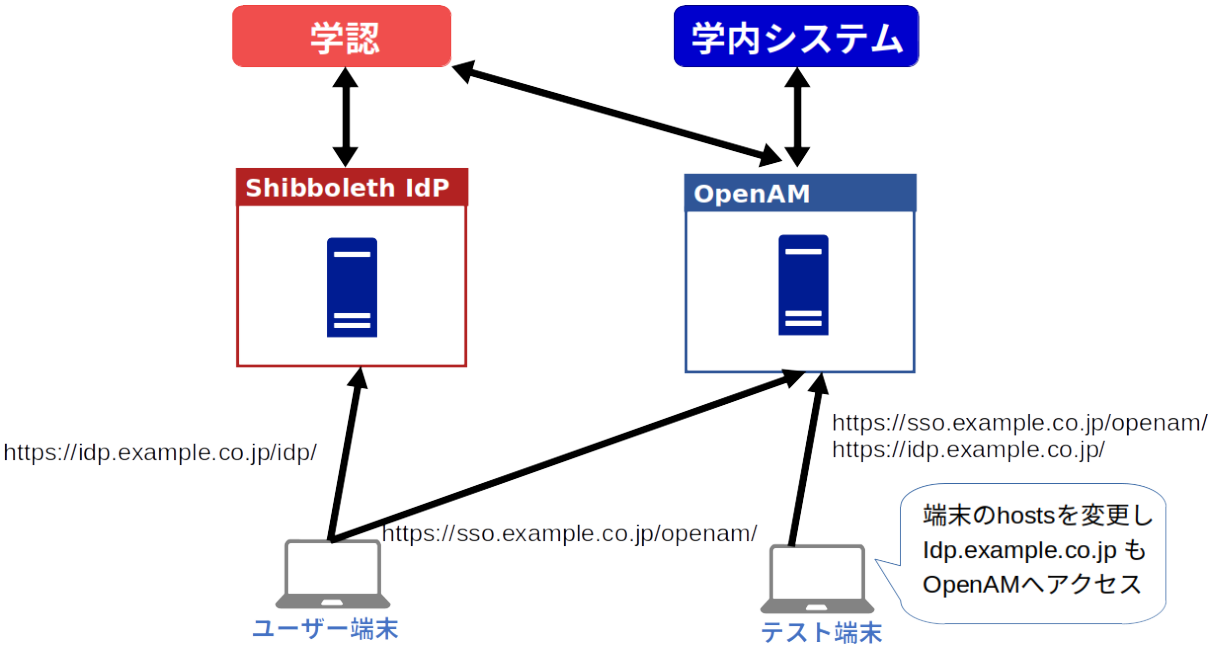

図 3　並行稼働の構成

このような並行稼働により、テスト端末から OpenAM を経由した学認参加で問題ないことを確認してから、本格的に切り替えることが可能です。

図では示しておりませんが、Shibboleth IdP と OpenAM は同じユーザーレポジトリ (OpenLDAP) を参照している前提です。 また、Cookie を共有する必要があるため、OpenAM のアクセス FQDN と Shibboleth IdP のアクセス FQDN とでドメインが一致している必要があります。

2 OpenAM の設定

本章では Shibboleth IdP から OpenAM への移行において必要な設定を説明します。

2.1 Shibboleth IdP の FQDN を OpenAM に登録

OpenAM に対して Shibboleth IdP の FQDN でもアクセス出来るよう、FQDN マップの設定を行います。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「設定」 -> 「デフォルトサーバー」-> 「詳細設定」と辿ります。
3. 画面の一番下より次の値を入力します。
   - プロパティ名に com.sun.identity.server.fqdnMap[【Shibboleth IdP の FQDN】]
   - プロパティ値に 【Shibboleth IdP の FQDN】

   Shibboleth IdP の FQDN が idp.example.co.jp の場合の設定値は次のとおりです。
   プロパティ名: com.sun.identity.server.fqdnMap[idp.example.co.jp]
   プロパティ値: idp.example.co.jp

4. ＋を押して、値が追加されたことを確認し「変更の保存」を押します。

続けて、レルムの「レルムまたは DNS のエイリアス」に【Shibboleth IdP の FQDN】を登録します。

1. 上部メニューのレルムから対象のレルムをクリックします。
2. レルムの概要の右側の 「プロパティ」を押します。
3. 「レルムまたは DNS のエイリアス」に【Shibboleth IdP の FQDN】を追加し「変更の保存」を押します。

以上で設定完了です。この設定が完了すると 【Shibboleth IdP の FQDN】で OpenAM にアクセスすることが出来るようになります。

2.2 OpenAM の旧監査ログ設定

SAML に関するログを記録するため、OpenAM の旧監査ログを出力します。

1. OpenAM に管理者ユーザーでログインします。
2. 上部メニューの「設定」 -> 「グローバルサービス」-> 「システム」-> 「ログ」と辿ります。
3. 下記の設定を行います。

設定項目	設定値
ログ状態	アクティブ
ログローテーション	有効のチェックを外す

旧監査ログのログローテートを logrotate で行います。具体的には、/opt/osstech/etc/logrotate.d/openam に旧監査ログのローテート設定を追加します。

- /opt/osstech/etc/logrotate.d/openam

```
/opt/osstech/var/lib/tomcat/data/openam/openam/stats/*
/opt/osstech/var/lib/tomcat/data/openam/openam/debug/*
/opt/osstech/var/lib/tomcat/data/openam/openam/log/*.access    <- 追加
/opt/osstech/var/lib/tomcat/data/openam/openam/log/*.error     <- 追加
{
  rotate 100
  daily
...
```

設定の反映には Tomcat の再起動が必要ですが、再起動はこのあとの手順 (キーストアファイルの変更) で行います。

2.3 署名鍵/暗号鍵を OpenAM のキーストアへインポート

「PEM 形式のサーバー証明書」と「秘密鍵」を OpenAM のキーストアにインポートします。 Shibboleth IdP では idp.properties で指定しているファイルです。 学認が公開している手順に従って Shibboleth IdP を構築していた場合、署名鍵と暗号鍵は同じファイルを利用しています。

```
idp.signing.key= %{idp.home}/credentials/server.key
idp.signing.cert= %{idp.home}/credentials/server.crt
idp.encryption.key= %{idp.home}/credentials/server.key
idp.encryption.cert= %{idp.home}/credentials/server.crt
```

「PEM 形式のサーバー証明書」(server.crt) と「秘密鍵」(server.key) を OpenAM サーバー上に配置してください。(本書では/tmp に配置したと仮定します。)

これらのファイルを、keytool コマンドを利用し、OpenAM のキーストアにインポートします。 OpenAM が利用しているキーストアファイルは、下記の設定項目で指定しているファイルです。

- キーストアの設定箇所
  - 「設定」タブ → 「デフォルトサーバー」 → 「セキュリティ」 →「キーストア」 → 「キーストアファイル」

本書ではキーストアファイルは/opt/osstech/var/lib/data/openam/private/mykeystore.jceks としますので、自身の環境に合わせて読み替えてください。

キーストアに格納するエイリアス名を決める必要があります。エイリアス名は学認用であると判別出来るように GakuNin の文字を入れ、証明書の更新に備えて発行した年を含めることを推奨します。本書ではエイリアス名を gakunin-cert-2023 とします。最初に PEM 形式のサーバー証明書と秘密鍵を PKCS#12 形式に変換します。

```
# openssl pkcs12 \
    -export \
    -in /tmp/server.crt \
    -inkey /tmp/server.key \
    -out /tmp/server.pkcs12 \
    -name "gakunin-cert-2023"
```

実行するとパスワードを聞かれるため、任意のパスワードを入力します。

```
Enter Export Password:  <- 任意のパスワードを入力
Verifying - Enter Export Password:  <- 同じパスワードを入力
```

パスワード入力後、何も表示されずコマンドが終了すれば成功です。 作成した PKCS#12 形式のファイルをキーストアにインポートします。

```
# keytool -importkeystore \
    -srckeystore /tmp/server.pkcs12 \
    -destkeystore /opt/osstech/var/lib/data/openam/private/mykeystore.jceks \
    -srcstoretype pkcs12 \
    -srcalias "gakunin-cert-2023" \
    -deststoretype jceks \
    -destalias "gakunin-cert-2023" \
    -deststorepass "【キーストアパスワード】" \
    -destkeypass "【秘密鍵のパスワード】"
```

【キーストアパスワード】や【秘密鍵のパスワード】は、OpenAM キーストア作成時に設定したパスワードです。

コマンドを実行すると Enter source keystore password: と聞かれるため、PKCS#12 形式変換時に入力した任意のパスワードを入力します。

```
Enter source keystore password:  <- openssl pkcs12 実行時に入力したパスワード
```

パスワード入力後 Warning のメッセージが表示されることがありますが、問題はありませんので無視します。インポートが完了したら、キーストアに含まれているか確認します。

```
# keytool -list \
    -keystore /opt/osstech/var/lib/data/openam/private/mykeystore.jceks \
    -alias "gakunin-cert-2023" \
    -storepass  "【キーストアパスワード】"
```

コマンドを実行し、次のようにフィンガープリントなどが表示されれば成功です。

```
gakunin-cert-2023, Apr 20, 2023, PrivateKeyEntry,
Certificate fingerprint (SHA-256): XX:XX:XX:...
```

インポート作業終了後、PEM 形式や PKCS#12 形式のファイルは不要なため削除します。

```
# rm /tmp/server.key /tmp/server.crt /tmp/server.pkcs12
```

2.4 学認メタデータの署名検証用証明書のインポート

学認が提供するメタデータは署名されているため、この署名を検証するための署名用証明書を OpenAM のキーストアにインポートします。

まず、メタデータの署名用証明書を学認のサイトからダウンロード[*1]し OpenAM サーバーに配置します。 (本書では/tmp に配置したとします。)

```
# keytool -importcert \
    -alias gakunin-signer-2017 \
    -noprompt \
    -trustcacerts \
    -file /tmp/gakunin-signer-2017.cer \
    -keystore "/opt/osstech/var/lib/data/openam/private/mykeystore.jceks"
```

「Shibboleth IdP の署名鍵」と「学認メタデータの署名検証用証明書」のインポートが完了し、かつ OpenAM を複数台構成で構築している場合は、キーストアファイルを他のサーバーにコピーしてください。全ての OpenAM サーバーで同じファイルを使用します。

```
# scp /opt/osstech/var/lib/data/openam/private/mykeystore.jceks \
   [OpenAM 別号機]:/opt/osstech/var/lib/data/openam/private/mykeystore.jceks
```

キーストアの変更を反映させるため Tomcat を再起動します。

```
# systemctl restart osstech-tomcat
```

[*1] 運用フェデレーションとテストフェデレーションで署名検証用証明書が異なります。OpenAM が参加するフェデレーションの署名用証明書をダウンロードしてください。

2.5 ホスト IdP プロバイダーの作成

OpenAM 管理コンソールからホスト IdP プロバイダーを作成します。

1. OpenAM に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3.「共通タスク」で「SAMLv2 プロバイダを作成」をクリックします。
4.「SAMLv2 プロバイダを作成」のメニューから「ホストアイデンティティープロバイダの作成」をクリックします。
5.「このプロバイダのメタデータがありますか?」は「いいえ」をチェックします。
6.「メタデータ」の「名前」の URL を Shibboleth IdP の EntityID に変更します。
7.「署名鍵」はプルダウンメニューから選択します。「署名鍵/暗号鍵を OpenAM のキーストアへインポート」で指定したエイリアス名を選択してください。
8.「新しいトラストサークル」にトラストサークルの名前を入力します。ここでは”GakuNin” と入力します。
9.「ベース URL」の URL を、一般ユーザーが OpenAM へアクセスする URL(本書では https://sso.example.co.jp/openam) に変更します。
10. 画面右上の「設定」ボタンをクリックして設定を保存します。属性マッピングは特に設定しません。
11. 設定完了の画面が表示されます。次のアクションを求められますが、ここでは「終了」ボタンをクリックします。

ホスト IdP プロバイダー作成が完了しました。続いて、作成したホスト IdP プロバイダーに学認に関連した設定を行います。

1. 画面上部のメニューから「連携」を押し、 エンティティープロバイダの一覧から「Shibboleth IdP の EntityID」を押します。
2.「表明コンテンツ」タブの「NameID の書式リスト」から「urn:oasis:names:tc:SAML:2.0:nameid-format:persistent」を削除し、画面右上の「保存」を押します。
3.「表明処理」タブのアカウントマッパーの「Disable NameID Persistence:」のチェックボックスにチェックを入れ、画面右上の「保存」を押します。
4.「表明処理」タブの属性定義用スクリプトで「SAML Attribute Resolution Script」を選択し、画面右上の「保存」を押します。
5. サービスタブの下記のエンドポイントの URL を Shibboleth IdP のメタデータに記載されている URL に変更し、画面右上の「保存」を押します。

```
[シングルサインオンサービス] - [HTTP-REDIRECT] - [場所]
-> Shibbleth IdP メタデータの SingleSignOnService Binding="urn:oasis:names:tc:SAML
   :2.0:bindings:HTTP-Redirect" の Location の URL

[シングルサインオンサービス] - [POST] - [場所]
-> Shibbleth IdP メタデータの SingleSignOnService Binding="urn:oasis:names:tc:SAML
   :2.0:bindings:HTTP-POST" の Location の URL
```

以上で、ホスト IdP プロバイダーの作成は完了です。

2.6 リモート SP プロバイダー初期値の設定

OpenAM 管理コンソールから、リモート SP プロバイダー初期値の設定を作成します。

1. OpenAM に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3. 左のサイドメニューの「サービス」を開き、「サービスの追加」を押します。
4. サービスタイプに「SAMLv2 リモート SP 初期値設定」を選択し、「作成」を押します。
5. 「属性送信の同意画面を表示する」を有効にし、「設定の保存」を押します。

2.7 メタデータ自動更新の設定

OpenAM 管理コンソールからメタデータ自動更新の設定を行い、学認のメタデータを取り込む設定を行います。

1. OpenAM に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3. 左のサイドメニューの「サービス」を開き、「サービスの追加」を押します。
4. サービスタイプに「SAMLv2 メタデータの自動更新」を選択し、「作成」を押します。
5. 設定画面となります。下記を設定し「変更の保存」を押します。
   - 「キー」に “学認のメタデータ取得 URL” をセットします。
   - 「値」はメタデータファイルの保存先のため、OpenAM 設定ディレクトリ配下 (例: /opt/osstech/var/lib/tomcat/data/openam/openam/gakunin-metadata.xml) を指定します。「キー」と「値」を入力したら「+add」を押します。
   - 「実行時刻」は取得を実行する時刻です。任意の時刻で構いません。例えば 04:00:00(午前 4 時に実行) と指定します。
   - 「対象とするロール」は “なし” とします。

- 「対象とするエンティティ」に利用する SP の EntityID を全て登録します。[*2]
- 「エンティティの新規登録を許可する」「有効期限をチェックする」「署名を検証する」は有効とします。

2.7.1 メタデータ読み込み

メタデータ自動更新の設定が終わったらメタデータの読み込みを行います。OpenAM の管理者ログインが完了している状態で下記の URL へアクセスします。[*3]

- https://【FQDN】/【コンテキストパス】/saml2/jsp/reload.jsp?realm=【レルム】

メタデータの読み込みに成功すると「Complete」と表示されます。もし「Complete」と表示されない場合は、設定に誤りや不足がありますので、設定や OpenAM のデバッグログ (MetadataReloadTaskFactory) を確認してください。

メタデータの読み込みは、OpenAM 管理者による認証済み状態でアクセスする必要があります。OpenAM 未ログイン状態でアクセスするとログイン画面にリダイレクトされます。一般ユーザーによる認証済み状態でアクセスすると `This action is only allowed for admin user.` と表示され、メタデータの読み込みは行われません。

[*2] OpenAM は、ここで指定した SP のみメタデータを取り込みます。 Shibboleth IdP の設定 (attribute-filter.xml) やログ (idp-audit.log) から、利用している学認 SP を確認して設定してください。

[*3] 例として、OpenAM セットアップガイドに従って構築した OpenAM の場合、メタデータ更新 URL は次のとおりです。https://openam01.example.co.jp/openam/saml2/jsp/reload.jsp?realm=/sso

2.8 学認で利用する属性の生成

Shibboleth IdP の attribute-resolver.xml にあたる設定を OpenAM に行います。

2.8.1 データストアのユーザー属性に学認用の属性を追加

LDAP のユーザーエントリにある学認用の属性を OpenAM で参照する設定を行います。Shibboleth IdP の設定から使用している LDAP の属性[*4]を確認し、OpenAM に設定します。

1. OpenAM に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3. 左のメニューの「データストア] を押し、「(データストア名)」と辿ります。
4. 「LDAP ユーザー属性」について、Shibboleth IdP で使用し「現在の値」のリストに含まれていない属性を追加します。
5. 「保存」を押します。

Shibboleth IdP で phonetic が付いた属性を利用している場合は、「LDAP ユーザー属性」では phonetic を付けた属性を設定します。

例:
`givenName;lang-ja` **を「LDAP ユーザー属性」のリストに追加する。**

2.8.2 SAML 属性送信のスクリプトを作成

Shibboleth IdP で LDAP の属性値をそのまま SP に送るのではなく何らかの加工をしていた場合は、 OpenAM の SAML 属性送信のスクリプトを使って生成します。

SAML 属性送信のスクリプトの編集は管理コンソールより行います。

1. OpenAM に管理者ユーザーでログインします。
2. OpenAM 管理コンソールで対象のレルムをクリックします。
3. 左メニューの「スクリプト」を押し、スクリプトの一覧から「SAML Attribute Resolution Script」をクリックします。
4. 「名前」などの設定は変更せず、「スクリプト」のコードを編集し、「変更の保存」を押します。

「SAML Attribute Resolution Script」は初期状態でスクリプトがセットされています。初

[*4] 例えば attribute-resolver.xml の DataConnector id="myLDAP" の exportAttributes に記載されている属性は、OpenAM への設定が必要です。

期状態のスクリプトの関数 (function の定義) はいくつかそのまま利用しますが、利用しないコードは削除してしまって問題ありません。

以下、attribute-resolver.xml で定義している属性の type ごとに、生成方法を説明します。

2.8.2.1 type="Static" で生成した値

学認の attribute-resolver.xml テンプレートでは、 o と jao などで大学名を設定しています。全 SP で同じ値を送信するケースに使います。

- attribute-resolver.xml の設定

```
    <DataConnector id="staticO" xsi:type="Static"
                   exportAttributes="o">
        <Attribute id="o">
            <Value>Test Organization</Value>
        </Attribute>
    </DataConnector>

    <DataConnector id="staticJaO" xsi:type="Static"
                   exportAttributes="jao">
        <Attribute id="jao">
            <Value>テスト大学</Value>
        </Attribute>
    </DataConnector>
```

OpenAM のスクリプトでは下記のように定義します。

```
setStaticData();

function setStaticData() {
    putEntry("o", "Test Organization");
    putEntry("jao", "テスト大学");
}
```

2.8.2.2 type="Scoped" で生成した値

学認の attribute-resolver.xml テンプレートでは、eduPersonScopedAffiliation や eduPersonPrincipalName を Scoped を使って生成するサンプルが記載されています。

- attribute-resolver.xml

```
    <AttributeDefinition xsi:type="Simple" id="eduPersonAffiliation">
        <InputDataConnector ref="myLDAP" attributeNames="eduPersonAffiliation" />
    </AttributeDefinition>

    <AttributeDefinition xsi:type="Scoped" id="eduPersonScopedAffiliation"
                                                          scope="%{idp.scope}">
        <InputAttributeDefinition ref="eduPersonAffiliation" />
    </AttributeDefinition>
```

この設定で eduPersonScopedAffiliation は、LDAP の属性 eduPersonAffiliation の値にスコープ (idp.properties で定義) を繋げた値となります。 OpenAM のスクリプトでは下記のように定義します。

- OpenAM のスクリプト定義

```
var scope = "osstech.co.jp";
seteduPersonAffiliation();

function setScopeValue(name, ldap_attr) {
    var ldap_values = identity.getAttribute(ldap_attr);
    if (ldap_values) {
        var set_values = new java.util.HashSet();
        var iter = ldap_values.iterator();
        while (iter.hasNext()) {
            set_values.add(iter.next() + "@" + scope);
        }
        putEntry(name, set_values);
    }
}

function seteduPersonAffiliation() {
   setScopeValue("eduPersonScopedAffiliation", "eduPersonAffiliation");
}
```

2.8.2.3 type="Mapped" で生成した値

学認の attribute-resolver.xml テンプレートでは、 isMemberOf を Mapped を使って生成するサンプルが記載されています。

- attribute-resolver.xml

```
<AttributeDefinition xsi:type="Mapped" id="isMemberOf">
    <InputDataConnector ref="myLDAP" attributeNames="uid" />
    <ValueMap>
        <ReturnValue>https://voplatform1.example.ac.jp/gr/FooGroup</ReturnValue>
        <SourceValue>user0001</SourceValue>
    </ValueMap>
    <ValueMap>
        <ReturnValue>https://voplatform2.example.ac.jp/gr/FooGroup</ReturnValue>
        <SourceValue>user0002</SourceValue>
    </ValueMap>
    <ValueMap>
        <ReturnValue>https://voplatform99.example.ac.jp/gr/FooGroup</ReturnValue>
        <SourceValue>user0099</SourceValue>
    </ValueMap>
</AttributeDefinition>
```

この設定で isMemberOf は、LDAP の属性 uid の値によって値が変わります。uid が user0001,user0002,user0003 以外の値の場合は isMemberOf は SP に送信されません。 OpenAM のスクリプトでは次のように定義します。

- OpenAM のスクリプト定義

```
setisMemberOf();

function setisMemberOf(){
    var name = "isMemberOf";
    var ldap_values = identity.getAttribute("uid");
    if (ldap_values) {
        var set_values = new java.util.HashSet();
        var iter = ldap_values.iterator();
        while (iter.hasNext()) {
            var value = checkisMemberOf(iter.next());
            if (value) {
                set_values.add(value);
```

```
            }
        }
        putEntry(name, set_values);
    }
}

function checkisMemberOf(uid){
    if (uid == "user0001") {
        return "https://voplatform1.example.ac.jp/gr/FooGroup";
    }
    if (uid == "user0002") {
        return "https://voplatform2.example.ac.jp/gr/FooGroup";
    }
    if (uid == "user0099") {
        return "https://voplatform99.example.ac.jp/gr/FooGroup";
    }
}
```

2.8.2.4 type="SAML2NameID" で生成した値 (computedID)

学認の attribute-resolver.xml テンプレートでは、eduPersonTargetedID を type="SAML2NameID" を使い生成しています。

- attribute-resolver.xml

```
<AttributeDefinition xsi:type="SAML2NameID" id="eduPersonTargetedID"
            nameIdFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">
    <InputDataConnector ref="computedID" attributeNames="computedID" />
</AttributeDefinition>

<DataConnector id="computedID" xsi:type="ComputedId"
               excludeResolutionPhases="c14n/attribute"
               generatedAttributeID="computedID"
               salt="%{idp.persistentId.salt}"
               algorithm="%{idp.persistentId.algorithm:SHA}"
               encoding="%{idp.persistentId.encoding:BASE64}">
    <InputDataConnector ref="myLDAP"
                        attributeNames="%{idp.persistentId.sourceAttribute}" />
</DataConnector>
```

Shibboleth IdP では、 saml-nameid.properties でソルトやアルゴリズム等を設定します。OpenAM では、スクリプトの初期値として computedId を生成する setComputedID 関数を

用意しています。 この関数に Shibboleth IdP の設定内容を設定します。

項目	Shibboleth IdP のプロパティの設定項目名
ソースとなる属性	idp.persistentId.sourceAttribute
ソルト文字列	idp.persistentId.salt(secrets.properties で設定)
アルゴリズム	idp.persistentId.algorithm
エンコード方式	idp.persistentId.encoding

- OpenAM のスクリプト定義

```
setComputedID();

function setComputedID() {
    var attrs = identity.getAttribute("uid");
    if (attrs) {
        var uid = attrs.iterator().next();
        // Salt must be at least 16 bytes
        var computedId = utils.generateComputedId(uid, "BASE64", "SHA-1",
                                                  "VfIzEzattecXS97Obxkl");
        var nameId = utils.generateNameIdXml(computedId,
                 "urn:oasis:names:tc:SAML:2.0:nameid-format:persistent");
        putEntry("computedId", nameId);
    }
}
```

上記のスクリプト定義は「ソースとなる属性」が uid、「ソルト文字列」が VfIzEzattecXS97Obxkl、 「アルゴリズム」が SHA-1、「エンコード方式」が BASE64 となります。Shibboleth IdP の設定値に従って置き換えてください。

2.9 学認で利用する属性の送信設定

学認で利用する属性の生成で準備した属性について、SP 単位で属性を送信する設定を行います。Shibboleth IdP の attribute-filter.xml にあたる設定です。

基本的な設定方法は次のとおりです。

1. OpenAM に管理者ユーザーでログインします。
2. 画面上部のメニューから「連携」を押し、 エンティティープロバイダの一覧から「設定する SP の EntityID」を押します。
3. 「表明処理」タブの「属性マップ」[*5]に、SP に送信する属性を全て定義します。 「属性マップ」に設定する形式は [SAML 属性名]=[属性名] です。
4. eduPersonTargetedID を SP に送信する場合のみ、「エスケープしない属性」を設定します。(後述)
5. 画面右上の「保存」を押します。
6. SP の数だけ繰り返します。

[SAML 属性名] について説明します。学認の属性については、運用ガイドで SAML の属性名が定められております。 下記表に示しますので、 SP に送信する学認の属性としては、対応した SAML の属性名を設定します。

学認の属性名	SAML の属性名
eduPersonPrincipalName	urn:oid:1.3.6.1.4.1.5923.1.1.1.6
eduPersonTargetedID	urn:oid:1.3.6.1.4.1.5923.1.1.1.10
o(organizationName)	urn:oid:2.5.4.10
jao(jaOrganizationName)	urn:oid:1.3.6.1.4.1.32264.1.1.4
ou(organizationalUnitName)	urn:oid:2.5.4.11
jaou(jaOrganizationalUnitName)	urn:oid:1.3.6.1.4.1.32264.1.1.5
eduPersonAffiliation	urn:oid:1.3.6.1.4.1.5923.1.1.1.1
eduPersonScopedAffiliation	urn:oid:1.3.6.1.4.1.5923.1.1.1.9
eduPersonEntitlement	urn:oid:1.3.6.1.4.1.5923.1.1.1.7
mail	urn:oid:0.9.2342.19200300.100.1.3
givenName	urn:oid:2.5.4.42

[*5] 「属性マップ」については、SAML 設定ガイドの「NameID や属性で連携する値について」の説明を参照ください。

学認の属性名	SAML の属性名
jaGivenName	urn:oid:1.3.6.1.4.1.32264.1.1.2
sn	urn:oid:2.5.4.4
jasn	urn:oid:1.3.6.1.4.1.32264.1.1.1
displayName	urn:oid:2.16.840.1.113730.3.1.241
jadisplayName	urn:oid:1.3.6.1.4.1.32264.1.1.3
gakuninScopedPersonalUniqueCode	urn:oid:1.3.6.1.4.1.32264.1.1.6
isMemberOf	urn:oid:1.3.6.1.4.1.5923.1.5.1.1
eduPersonAssurance	urn:oid:1.3.6.1.4.1.5923.1.1.1.11
eduPersonUniqueId	urn:oid:1.3.6.1.4.1.5923.1.1.1.13
eduPersonOrcid	urn:oid:1.3.6.1.4.1.5923.1.1.1.16

「属性マップ」の設定例を示します。下記の 2 つの「属性マップ」を設定すると、SP には givenName と jaGivenName が連携されます。

```
urn:oid:2.5.4.42=givenName
urn:oid:1.3.6.1.4.1.32264.1.1.2=givenName;lang-ja
```

2.9.1 eduPersonTargetedID の設定について

eduPersonTargetedID はスクリプトで生成します。 スクリプトで生成した computedID の「表明処理」の設定は、次の 2 つが必要です。

- 「属性マップ」を次のように設定する。[*6]

```
urn:oasis:names:tc:SAML:2.0:attrname-format:uri|urn:oid:1.3.6.1.4.1.5923
.1.1.1.10=computedId
```

- 「エスケープしない属性」に設定を追加する

```
urn:oid:1.3.6.1.4.1.5923.1.1.1.10
```

eduPersonTargetedID は SAML の NameID フォーマットの形式で送信する必要があるため、上記の 2 つの設定が必要となります。

[*6] 表示の都合上改行していますが、実際の設定時は改行せず 1 行で設定します。

2.10 送信属性同意機能の設定

SAML アサーション送付前に送信属性の同意画面を表示する機能に関して、設定を行います。SAML 設定ガイドの「送信属性同意機能の設定」の章にある下記の作業を実施します。

- 組織認証用鍵ペアの作成
- 組織認証用の証明書エイリアスの変更
- 同意が必要な属性の設定
- 監査ログ

SAML 設定ガイドの「送信属性同意機能の有効化」の作業は、SP メタデータ読み込み時に有効になるため不要です。

「同意が必要な属性の設定」に設定する文言は任意ですが、参考情報として Shibboleth IdP(v4.2.1) の属性情報同意画面で表示される文言を設定する値を示します。

- eduPersonAffiliation

```
urn:oid:1.3.6.1.4.1.5923.1.1.1.1|en|Affiliation
urn:oid:1.3.6.1.4.1.5923.1.1.1.1|職位
```

- jao

```
urn:oid:1.3.6.1.4.1.32264.1.1.4|en|Organization name (written in Japanese)
urn:oid:1.3.6.1.4.1.32264.1.1.4|所属機関名［日本語］
```

- jaDisplayName

```
urn:oid:1.3.6.1.4.1.32264.1.1.3|en|Display name (written in Japanese)
urn:oid:1.3.6.1.4.1.32264.1.1.3|表示名［日本語］
```

- givenName

```
urn:oid:2.5.4.42|en|Given name
urn:oid:2.5.4.42|名
```

- eduPersonPrincipalName

```
urn:oid:1.3.6.1.4.1.5923.1.1.1.6|en|Principal name
urn:oid:1.3.6.1.4.1.5923.1.1.1.6|プリンシパル ID
```

- eduPersonTargetedID

```
urn:oid:1.3.6.1.4.1.5923.1.1.1.10|en|Unique ID for each service provider
urn:oid:1.3.6.1.4.1.5923.1.1.1.10|サービス毎のユニーク ID
```

- ou

```
urn:oid:2.5.4.11|en|Organizational unit
urn:oid:2.5.4.11|機関内所属名
```

- jasn

```
urn:oid:1.3.6.1.4.1.32264.1.1.1|en|Surname (written in Japanese)
urn:oid:1.3.6.1.4.1.32264.1.1.1|姓 [日本語]
```

- o

```
urn:oid:2.5.4.10|en|Organization name
urn:oid:2.5.4.10|所属機関名
```

- displayName

```
urn:oid:2.16.840.1.113730.3.1.241|en|Display name
urn:oid:2.16.840.1.113730.3.1.241|表示名
```

- jaGivenName

```
urn:oid:1.3.6.1.4.1.32264.1.1.2|en|Given name (written in Japanese)
urn:oid:1.3.6.1.4.1.32264.1.1.2|名 [日本語]
```

- eduPersonScopedAffiliation

```
urn:oid:1.3.6.1.4.1.5923.1.1.1.9|en|Scoped affiliation
urn:oid:1.3.6.1.4.1.5923.1.1.1.9|スコープ付き職位
```

- jaou

```
urn:oid:1.3.6.1.4.1.32264.1.1.5|en|Organizational unit (written in Japanese)
urn:oid:1.3.6.1.4.1.32264.1.1.5|機関内所属名［日本語］
```

- gakuninScopedPersonalUniqueCode

```
urn:oid:1.3.6.1.4.1.32264.1.1.6|en|Personal unique code
urn:oid:1.3.6.1.4.1.32264.1.1.6|個人識別番号
```

- sn

```
urn:oid:2.5.4.4|en|Surname
urn:oid:2.5.4.4|姓
```

- eduPersonEntitlement

```
urn:oid:1.3.6.1.4.1.5923.1.1.1.7|en|Entitlement
urn:oid:1.3.6.1.4.1.5923.1.1.1.7|資格情報
```

- mail

```
urn:oid:0.9.2342.19200300.100.1.3|en|E-mail
urn:oid:0.9.2342.19200300.100.1.3|メールアドレス
```

- isMemberOf

```
urn:oid:1.3.6.1.4.1.5923.1.5.1.1|en|Group identifier
urn:oid:1.3.6.1.4.1.5923.1.5.1.1|所属グループ識別子
```

- eduPersonAssurance

```
urn:oid:1.3.6.1.4.1.5923.1.1.1.11|en|Assurance level
urn:oid:1.3.6.1.4.1.5923.1.1.1.11|保証レベル
```

- eduPersonUniqueId

```
urn:oid:1.3.6.1.4.1.5923.1.1.1.13|en|Unique ID
urn:oid:1.3.6.1.4.1.5923.1.1.1.13|ユニーク ID
```

- eduPersonOrcid

```
urn:oid:1.3.6.1.4.1.5923.1.1.1.16|en|ORCID
urn:oid:1.3.6.1.4.1.5923.1.1.1.16|ORCID
```

2.11 トラストサークルの設定

トラストサークルに学認の SP を追加し、信頼関係を構築します。

OpenAM 管理コンソールから [連携] を押し、ホストアイデンティティプロバイダ作成時に作ったトラストサークルに IdP や学認 SP を追加します。

1. OpenAM に管理者ユーザーでログインします。
2. 画面上部のメニューから「連携」を押します。
3. ホスト IdP プロバイダーの作成で作成したトラストサークル名 (本書では GakuNin) を押します。
4. 「エンティティープロバイダ」の「選択可能」の一覧から学認の SP の EntityID を追加し、「保存」を押します。

2.12 Apache の設定

OpenAM サーバーにて Shibboleth IdP のエンドポイントのリクエストを受け付けるために、 サーバー証明書の設定や OpenAM の URL に Rewrite する設定を行います。サーバー証明書 (必要に応じて中間証明書) と秘密鍵は、Shibboleth IdP サーバーのファイルを使用します。

```
<VirtualHost *:443>
    ServerName 【Shibboleth IdP の FQDN】
    SSLCertificateKeyFile /etc/pki/tls/private/server.key
```

```
    SSLCertificateFile /etc/pki/tls/certs/server.crt
    SSLCertificateChainFile /etc/pki/tls/certs/chain.crt

    RewriteEngine on
    RewriteRule ^/idp/profile/SAML2/Redirect/SSO
                  /openam/SSORedirect/metaAlias/【メタエイリアス】 [PT]
    RewriteRule ^/idp/profile/SAML2/POST/SSO
                  /openam/SSOPOST/metaAlias/【メタエイリアス】 [PT]
</VirtualHost>
```

2 つの RewriteRule の設定は表示の都合上改行していますが、実際の設定時は改行せずにそれぞれ 1 行で設定します。

【メタエイリアス】は OpenAM の管理コンソールにて確認出来ます。

- [連携] - [ホスト IdP の EntityID] - [サービス] タブ - [MetaAlias]

作業は以上で完了です。ここまでの作業を終えたら OpenAM は学認に参加出来ます。

3 運用ガイド

本章では、 OpenAM を学認に参加させて運用する際に必要となる作業について説明します。

3.1 SP の追加

利用する学認の SP を追加したい場合の手順について説明します。

作業実施前に、新しく追加する SP の 「EntityID」と「送信する属性」を確認します。 情報は Shibboleth IdP の attribute-filter.xml の設定内容から判断出来ます。通常は、学認サイトの SP 一覧ページの該当 SP の「IdP 管理者向け」に記載があります。

例として attribute-filter.xml の設定内容が下記の場合、「EntityID」 と「送信する属性」は次のとおりです。

- EntityID
  - https://www.elgaronline.com/oa/metadata
- 送信する属性
  - eduPersonScopedAffiliation

```
<!-- Policy for Elgaronline -->
<AttributeFilterPolicy id="PolicyforElgaronline">
    <PolicyRequirementRule xsi:type="Requester"
                value="https://www.elgaronline.com/oa/metadata" />

    <AttributeRule attributeID="eduPersonScopedAffiliation">
        <PermitValueRule xsi:type="ANY" />
    </AttributeRule>
</AttributeFilterPolicy>
```

次ページより、実際の手順を説明します。本書では次の SP を追加する手順を示します。

- EntityID
  - https://test-sp1.gakunin.nii.ac.jp/shibboleth-sp
- 送信する属性
  - eduPersonTargetedID
  - mail
  - displayName

3.1.1 該当 SP をメタデータ更新対象に追加

追加する SP のメタデータを取り込む設定を行います。

- OpenAM に管理者ユーザーでログインします。

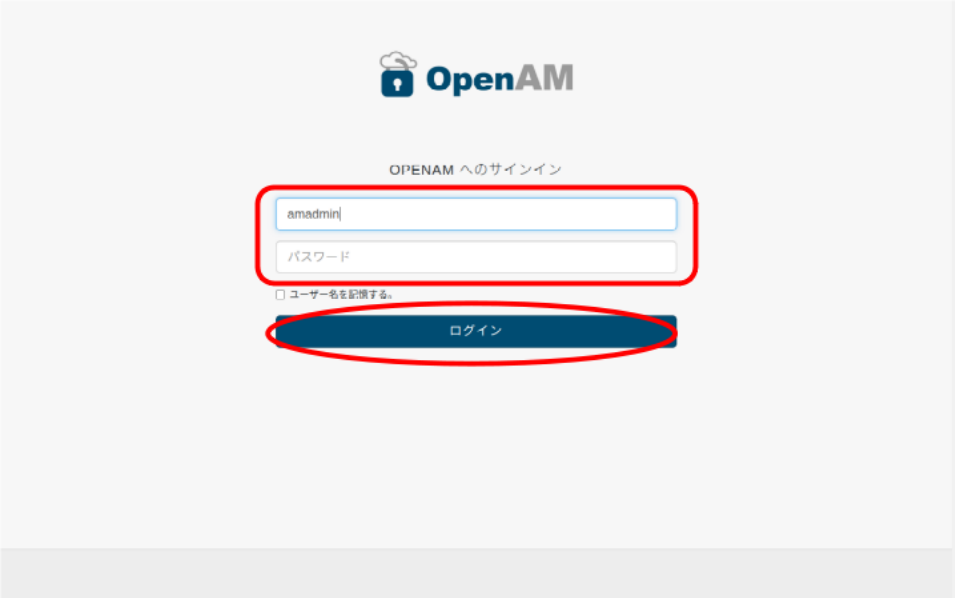

図 4　管理ユーザーログイン画面

- OpenAM 管理コンソールで対象のレルムをクリックします。(下図では demo レルム)

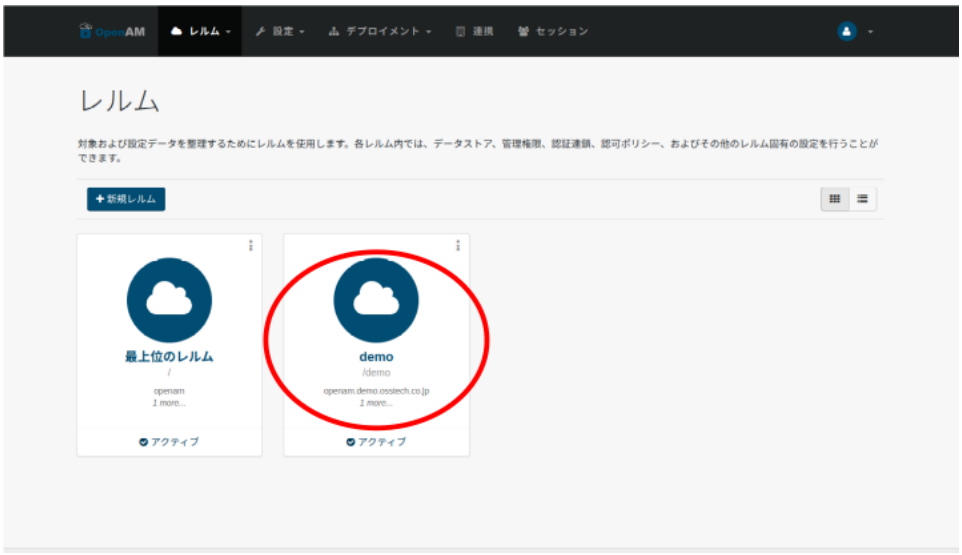

図 5　レルムの選択

- 左のサイドメニューの「サービス」を開きます。

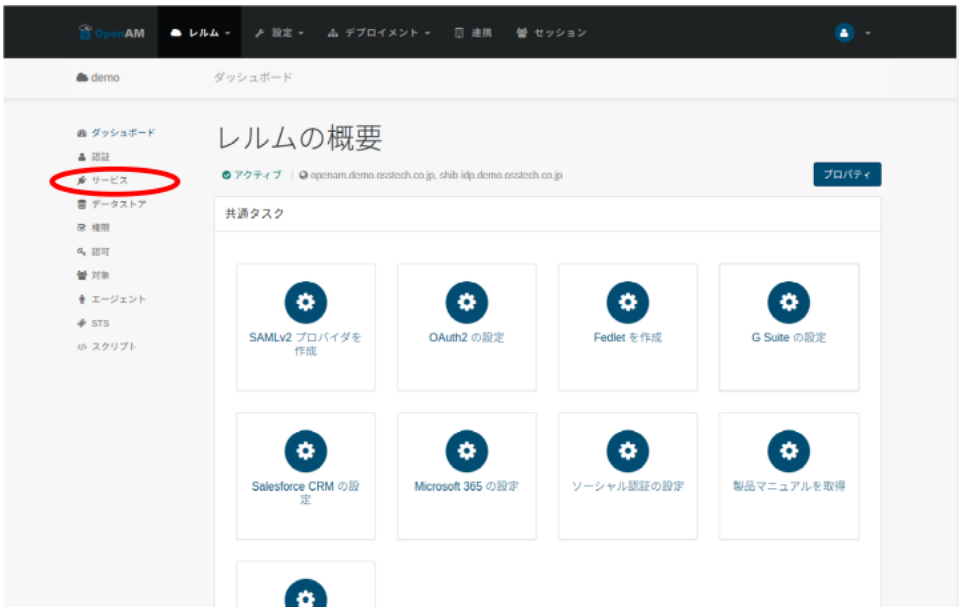

図 6　「サービス」を開く

*「SAMLv2 メタデータの自動更新」をクリックします。

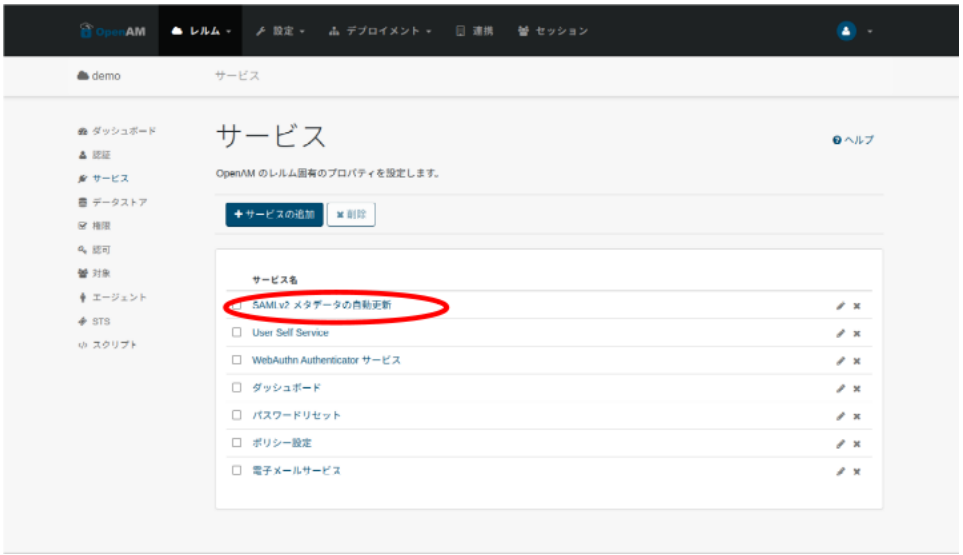

図 7　SAMLv2 メタデータの自動更新

- 「対象とするエンティティ」に追加する SP の EntityID を入力し

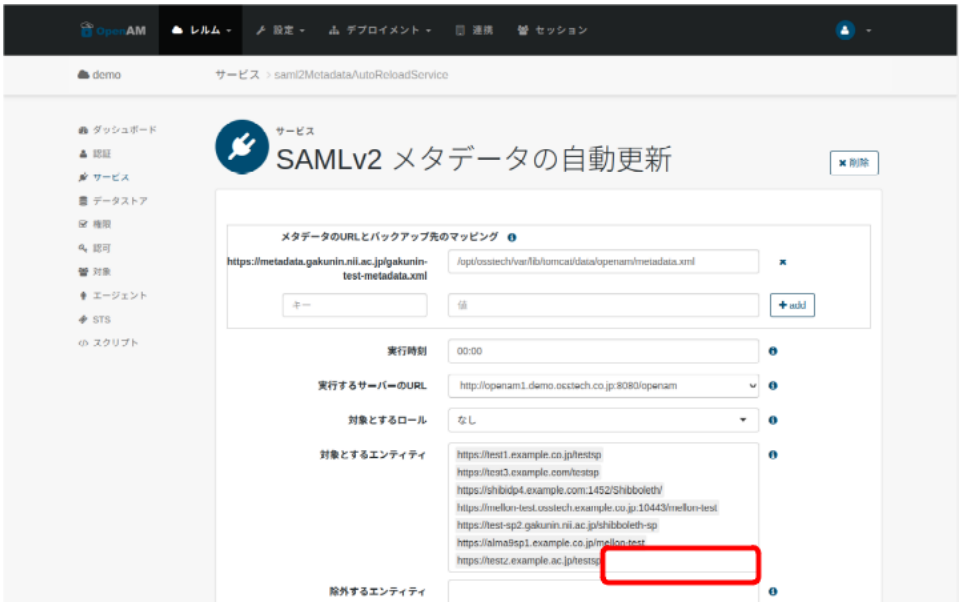

図 8　SP の EntityID 入力

- Add を押します。

図 9　SP の EntityID の追加（Add）

- 「対象とするエンティティ」に SP の EntityID が追加されたことを確認し、「変更の保存」を押します。

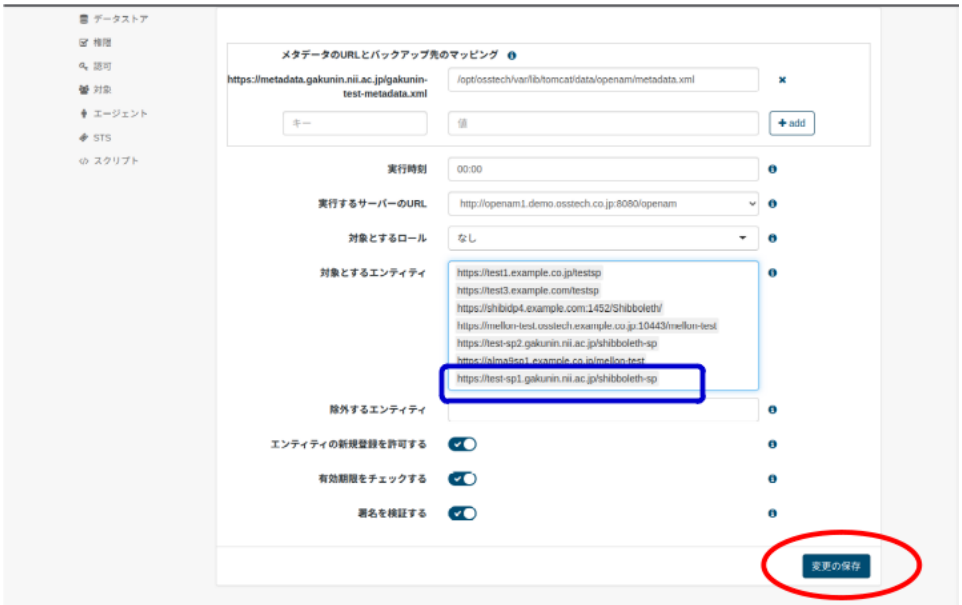

図 10　変更の保存

- 「変更を保存しました」と表示されれば完了です。

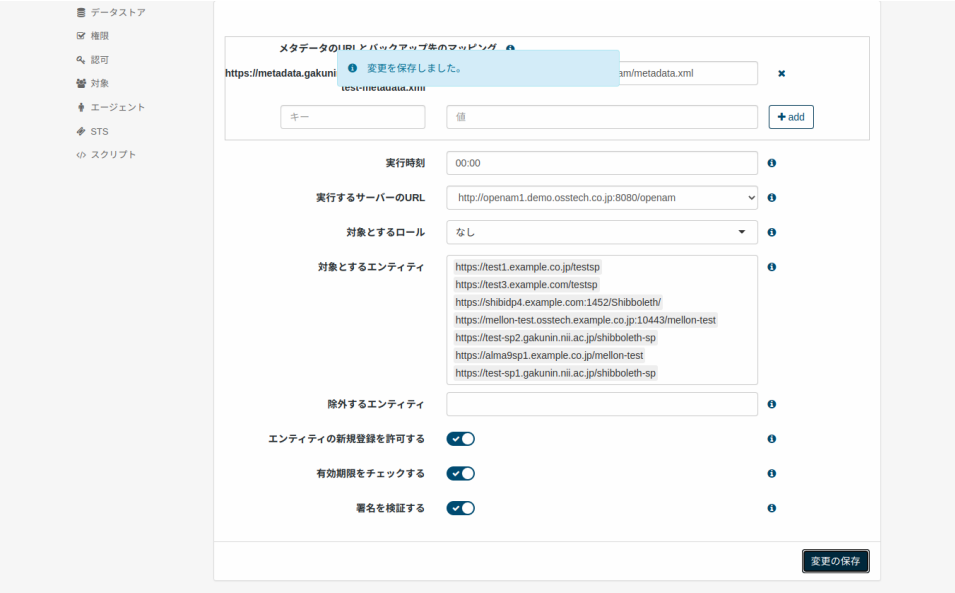

図 11　変更の保存の完了

3.1.2 メタデータの手動更新

メタデータ読み込みの手順に従ってメタデータ更新 URL にアクセスし、メタデータを更新します。

- OpenAM に管理者ユーザーでログインします。

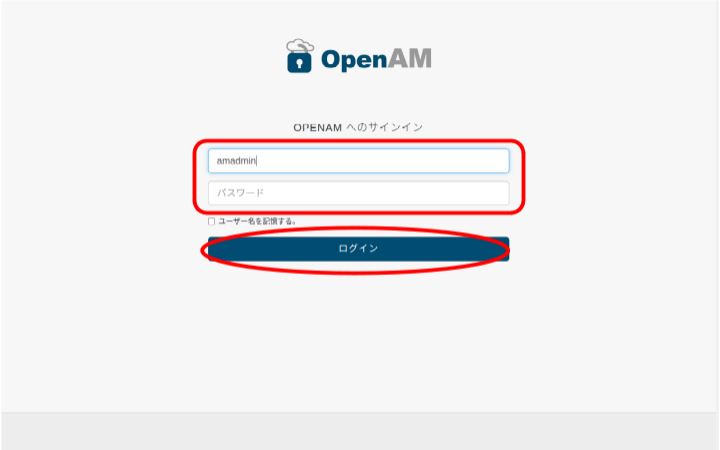

図 12　ログイン画面

- メタデータ更新 URL にアクセスし「Complete」と表示されるのを確認します。

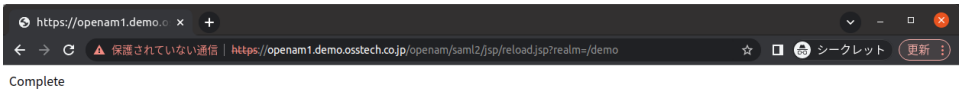

図 13　メタデータ更新 URL にアクセス

3.1.3 該当 SP をトラストサークルへ追加

トラストサークルの設定に従って、追加する SP の EntityID をトラストサークル名 (本書では GakuNin) に追加します。

- OpenAM に管理者ユーザーでログインします。

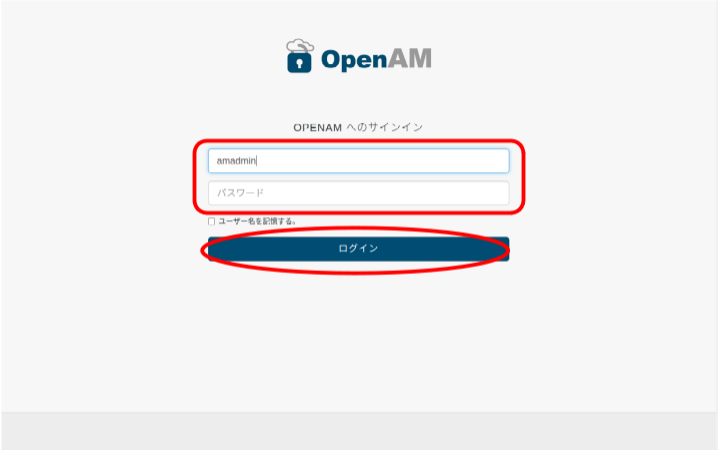

図 14　管理ユーザーログイン画面

- 画面上部のメニューから「連携」を押します。

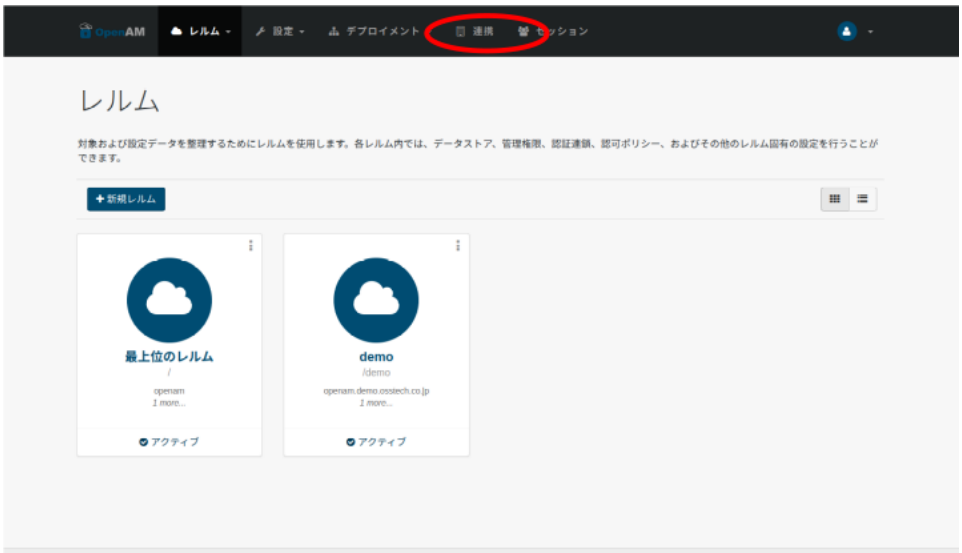

図 15　連携を押す

- トラストサークルの「GakuNin」を押します。

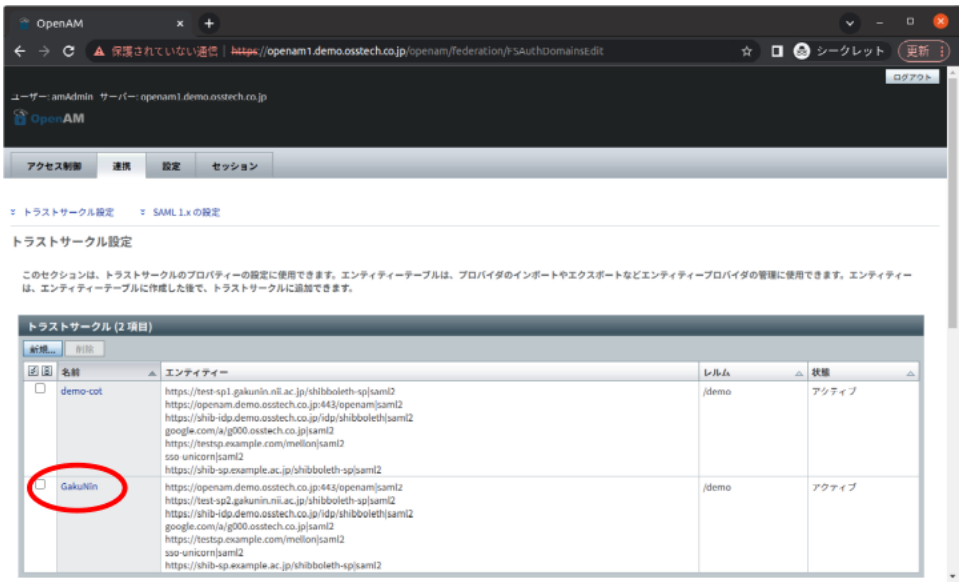

図 16　GakuNin を押す

- エンティティプロバイダの選択可能から「SP の EntityID」を選択し、「追加」を押します。

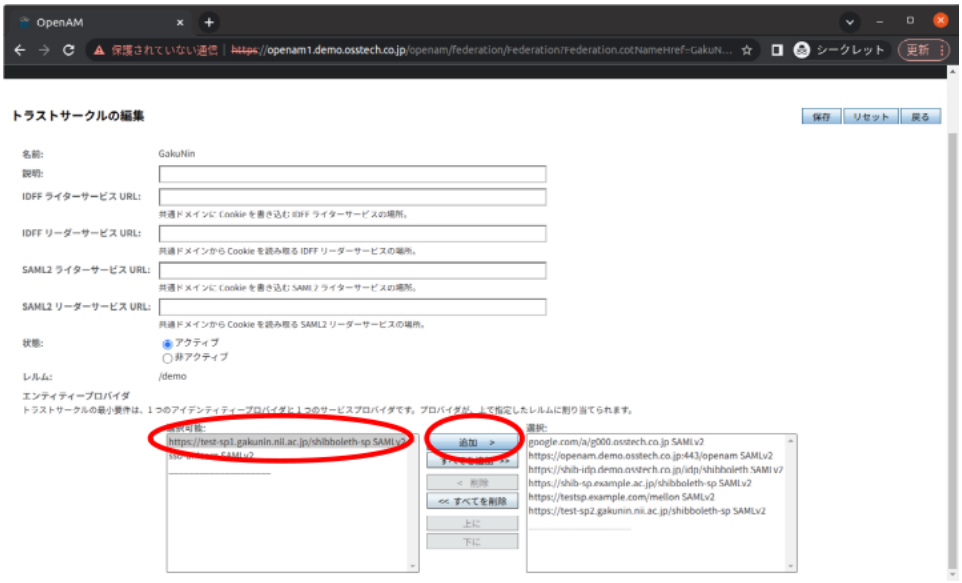

図 17　SP の EntityID を追加

- エンティティプロバイダの選択欄に SP の EntityID が追加されたことを確認し、「保存」を押します。

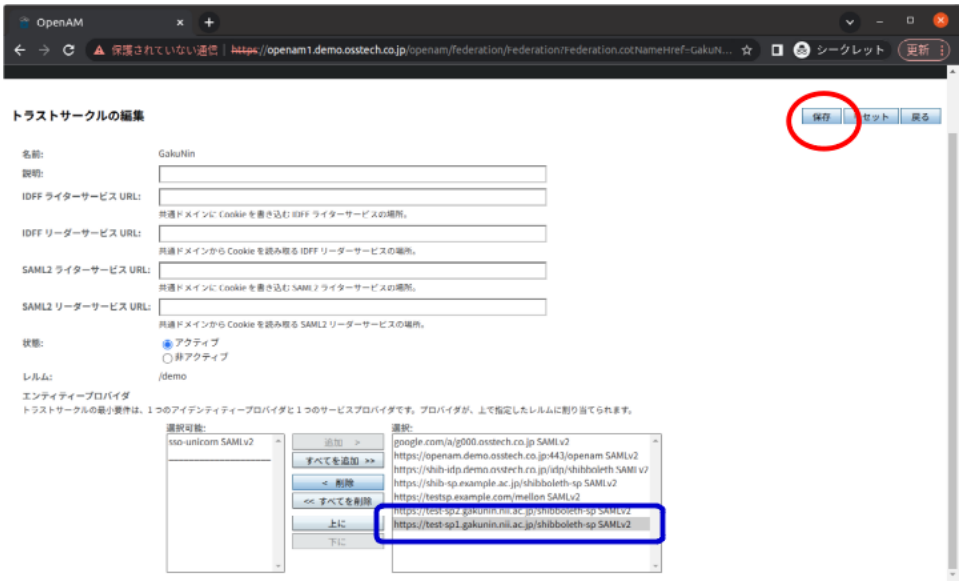

図 18　追加を確認し保存

- “トラストサークルプロファイルが更新されました。” と表示されれば完了です。

図 19　保存の確認

3.1.4 属性の設定

学認で利用する属性の送信設定に従って、追加する SP の送信属性の設定を行います。

- OpenAM に管理者ユーザーでログインします。

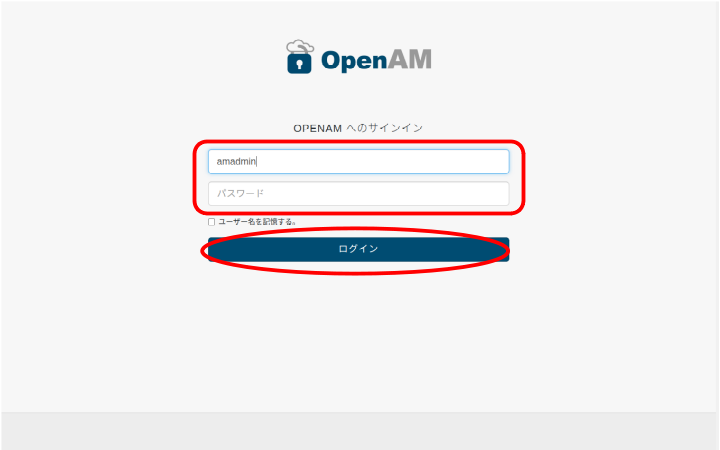

図 20　ログイン画面

- 画面上部のメニューから「連携」を押します。

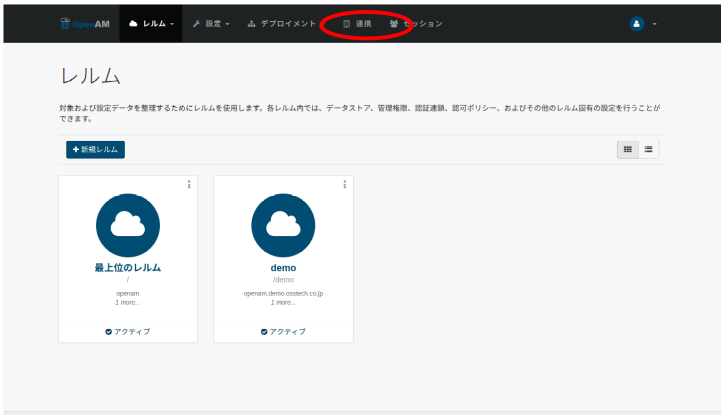

図 21　連携を押す

- エンティティプロバイダから「追加した SP の EntityID(本書では https://test-sp1.gakunin.nii.ac.jp/shibboleth-sp)」を押します。

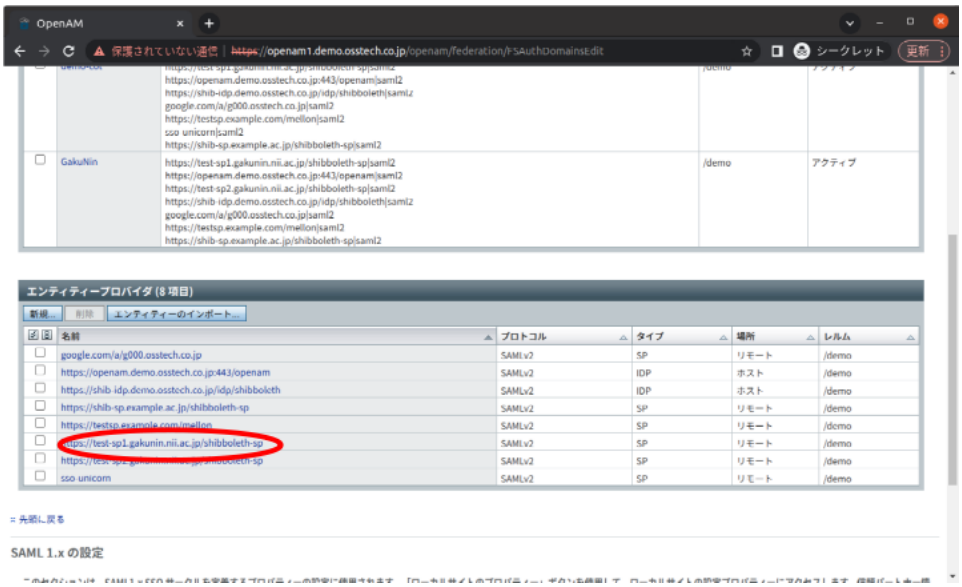

図 22　追加した SP の EntityID を押す

- 「表明処理」を押します。

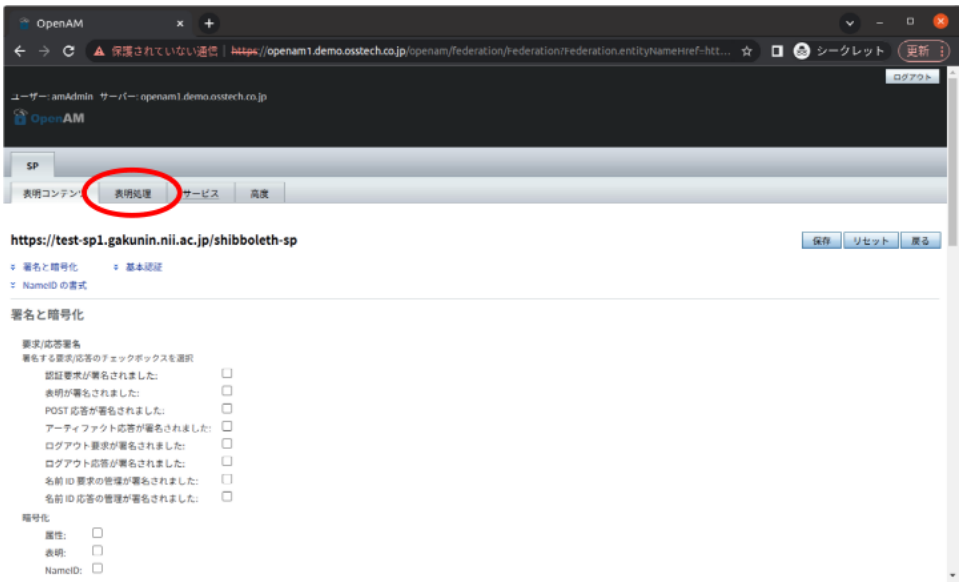

図 23　表明処理を押す

- 属性マップの「新しい値」に “[SAML 属性名]=[属性名]” を入力し、「追加」を押します。

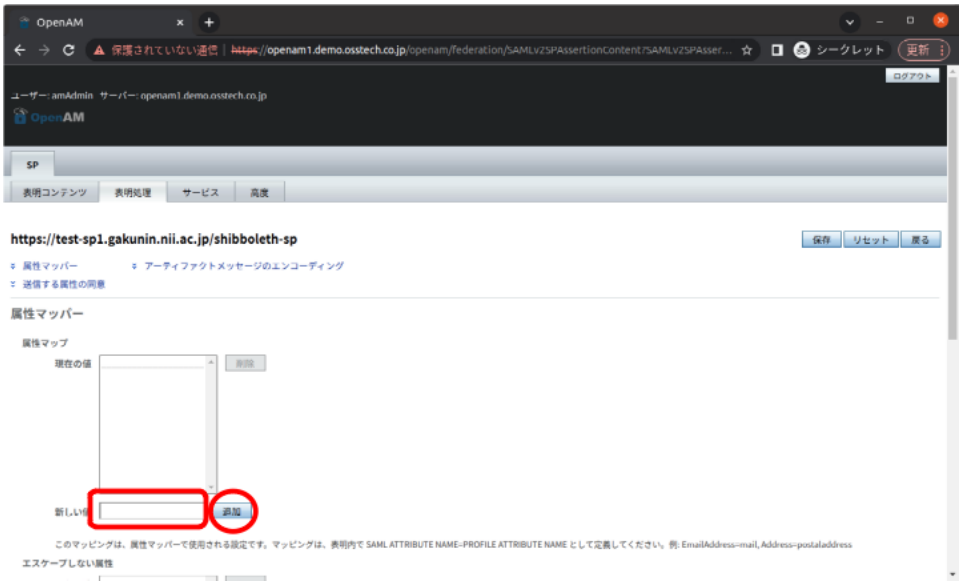

図 24　属性マップへ追加

- 現在の値に”[SAML 属性名]=[属性名]“が追加されていることを確認します。

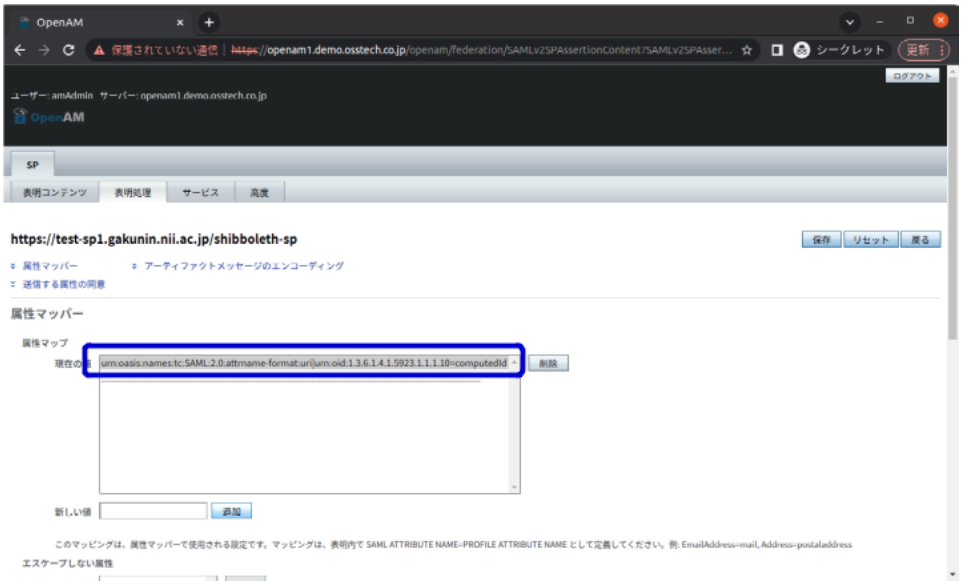

図 25　属性マップへ追加

- 送信する属性の数だけ設定を繰り返します。

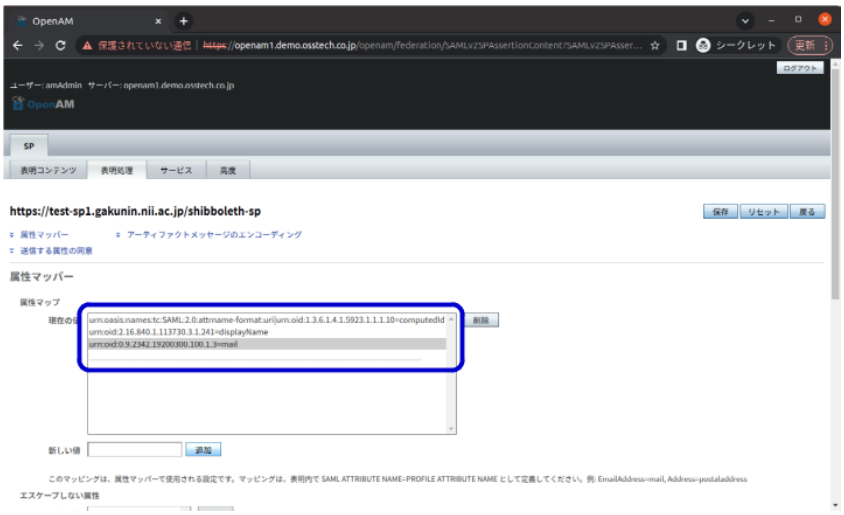

図 26　属性マップへ追加

- エスケープしない属性の「新しい値」に urn:oid:1.3.6.1.4.1.5923.1.1.1.10 を入力し、「追加」を押します。[*7]

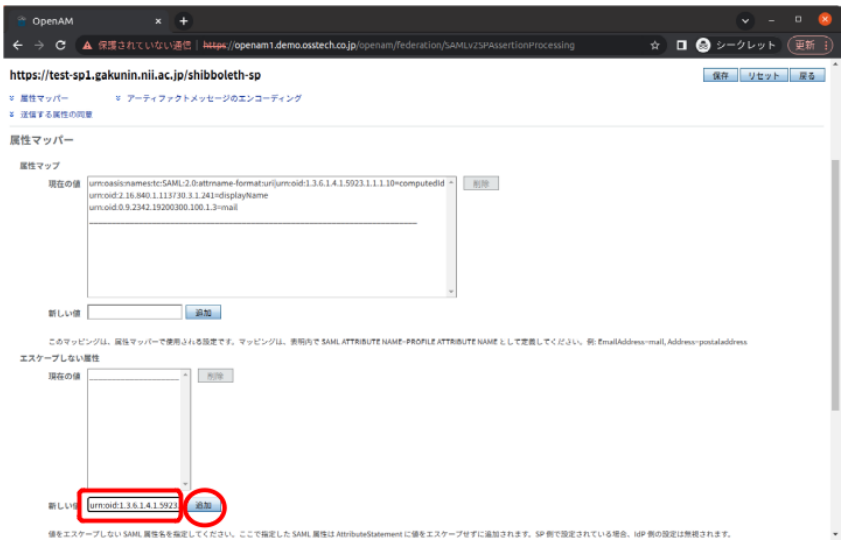

図 27　エスケープしない属性へ追加

[*7] エスケープしない属性の設定は、SP に eduPersonTargetedID を送信する場合のみ必要です。属性マップへ追加で eduPersonTargetedID を設定していない場合は、エスケープしない属性の設定は不要です。

- 現在の値に urn:oid:1.3.6.1.4.1.5923.1.1.1.10 が追加されていることを確認し、「保存」を押します。

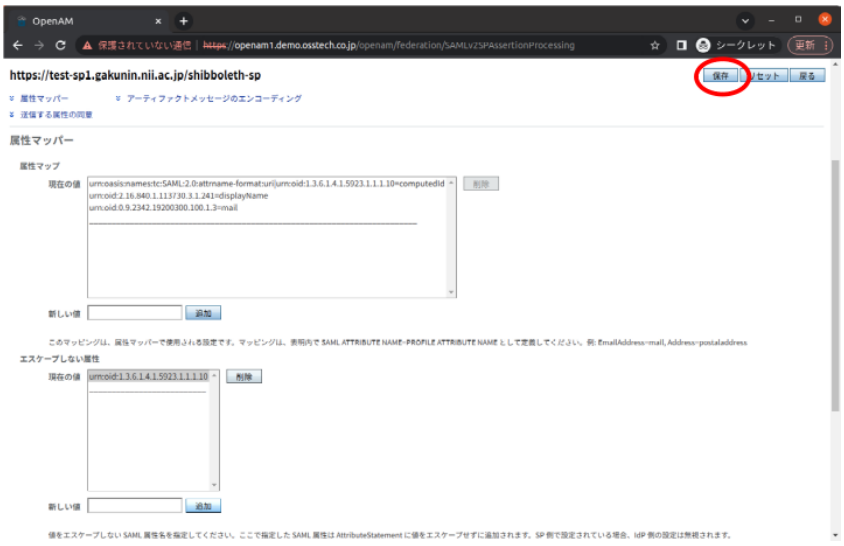

図 28　保存を押す

- “SAMLv2 サービスプロバイダプロパティーが更新されました。” と表示されれば完了です。

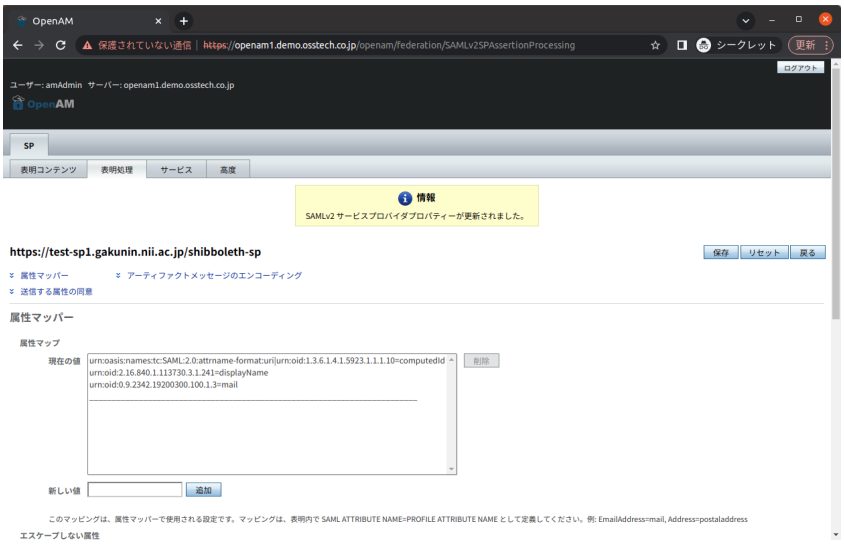

図 29　設定の更新確認

以上の作業で SP の追加作業は完了です。 追加した SP へアクセスし、サービスが利用可能なことを確認してください。

3.2 SP の削除

サービスを利用しなくなった等の理由により、SP を利用不可にする手順を説明します。
本章では次の SP を削除する手順を示します。

- EntityID
  - `https://test-sp1.gakunin.nii.ac.jp/shibboleth-sp`

3.2.1 該当 SP をトラストサークルから除外

利用しない SP をトラストサークルの一覧から削除することで、OpenAM の連携対象から外れます。

- OpenAM に管理者ユーザーでログインします。

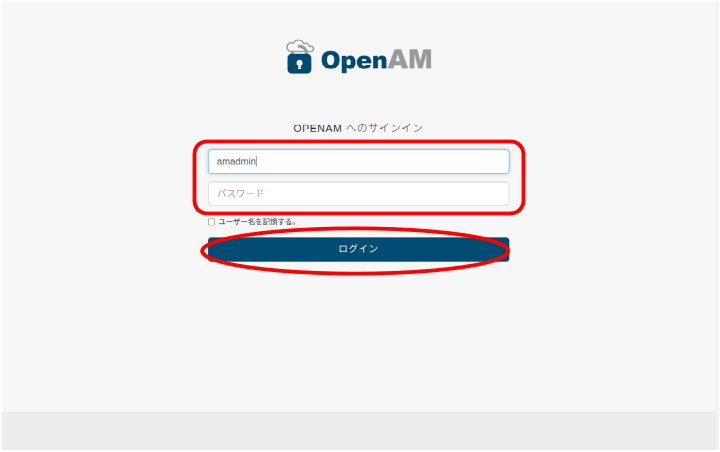

図 30　ログイン画面

- 画面上部のメニューから「連携」を押します。

図 31　連携を押す

- トラストサークルの「GakuNin」を押します。

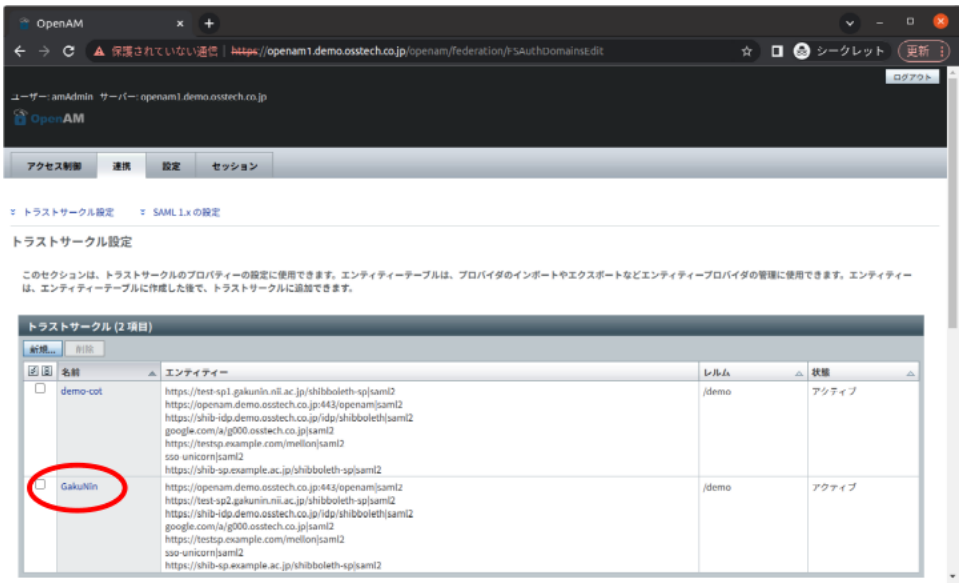

図 32　GakuNin を押す

- エンティティプロバイダの選択: から「SP の EntityID」を選択し、「削除」を押します。

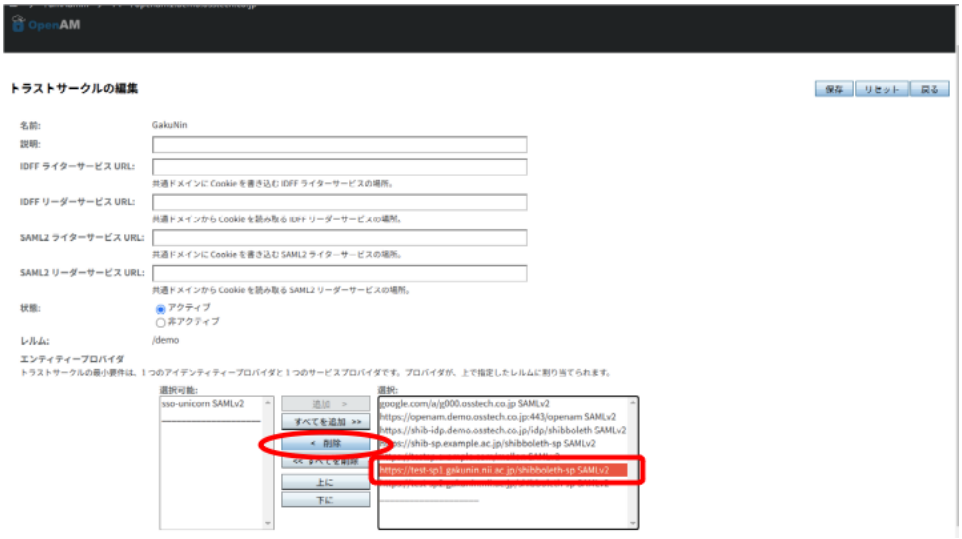

図 33　SP の EntityID を削除

- エンティティプロバイダの選択可能欄に SP の EntityID が追加されたことを確認し、「保存」を押します。

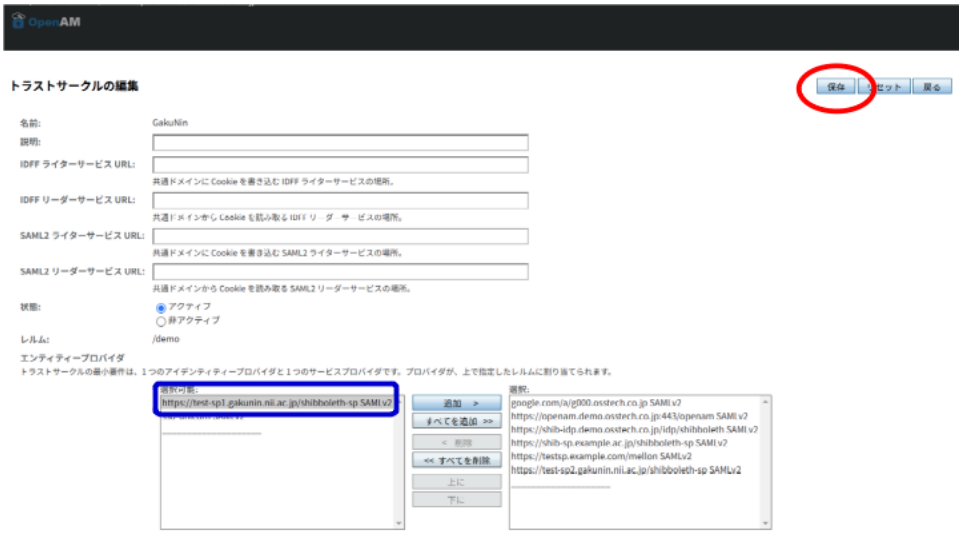

図 34　確認して保存

- “トラストサークルプロファイルが更新されました。” と表示されれば完了です。

図 35　保存の確認

SP を利用不可にする際は、トラストサークルからの除外のみを実施することを推奨します。利用者がトラストサークルの一覧に存在しない SP を利用しようとすると、OpenAM のエラー画面が表示されます。このように SP の設定自体は残しておくことで、今後改めて該当 SP の利用を再開する場合はトラストサークルの追加を実施することにより簡単に利用を再開できます。

もし今後利用する予定はなく SP 設定自体を削除したい場合は、メタデータ更新対象から除外と SP の設定の削除を実施します。

3.2.2 メタデータ更新対象から除外

利用しない SP について、メタデータを取り込まない設定を行います。

- OpenAM に管理者ユーザーでログインします。

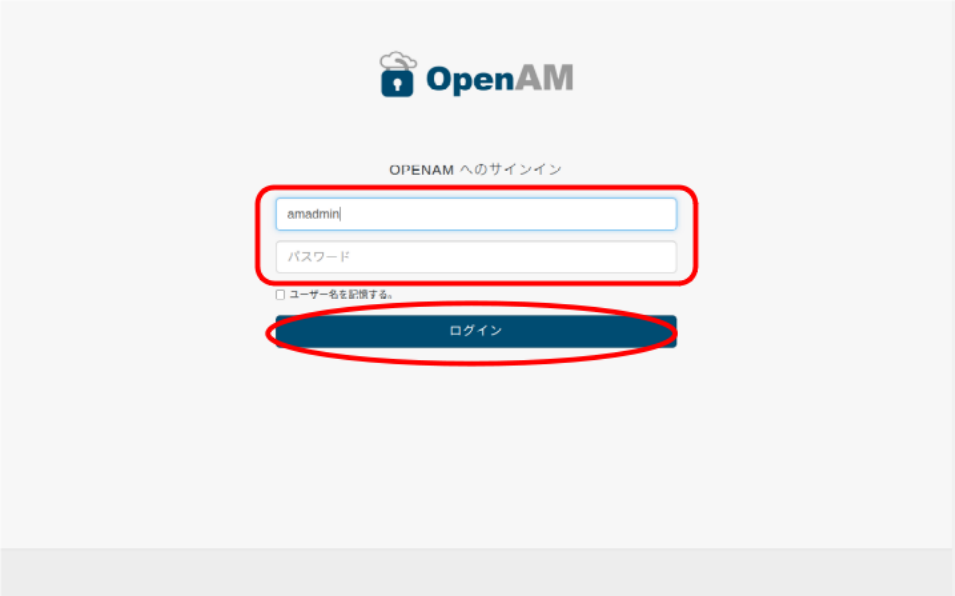

図 36　管理ユーザーログイン画面

- OpenAM 管理コンソールで対象のレルムをクリックします。(下図では demo レルム)

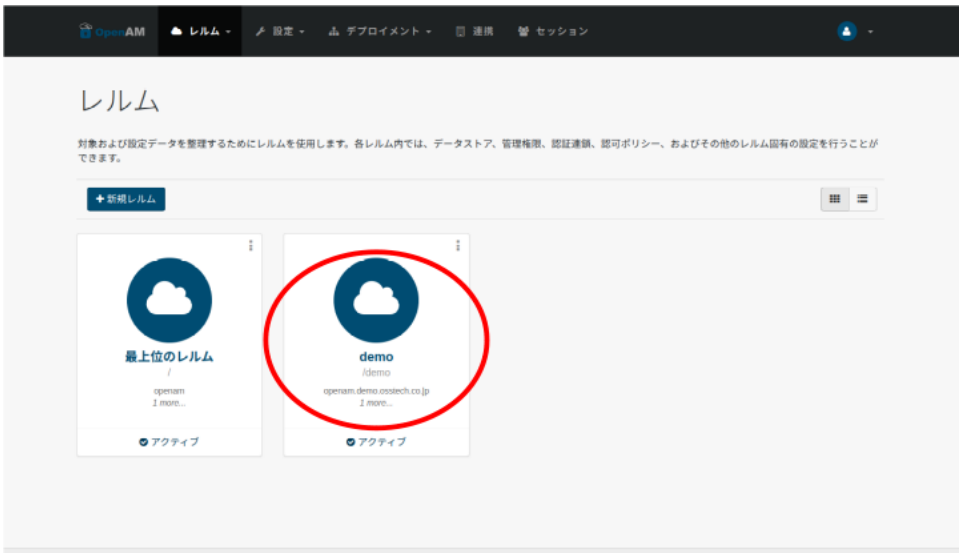

図 37　レルムの選択

- 左のサイドメニューの「サービス」を開きます。

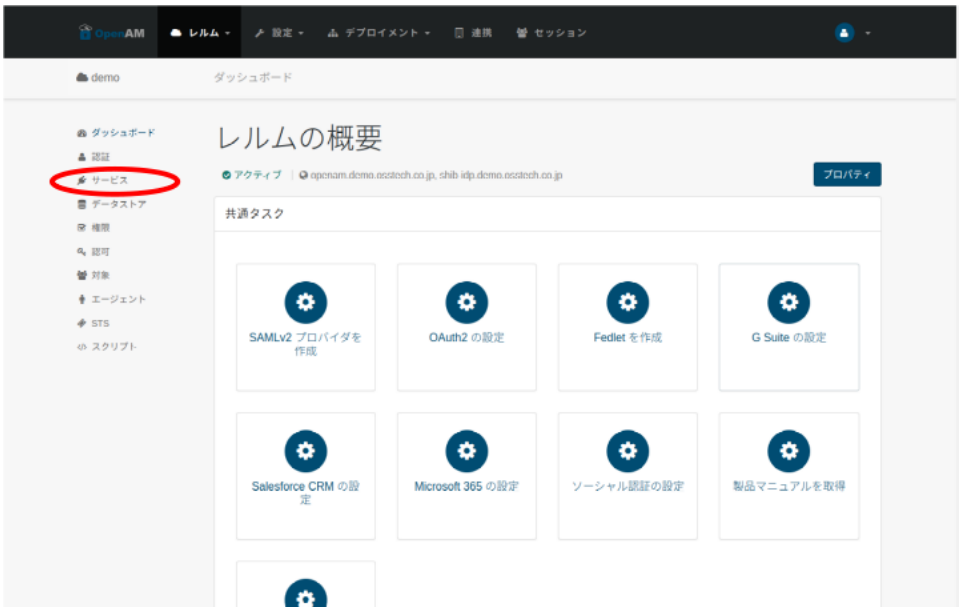

図 38　「サービス」を開く

*「SAMLv2 メタデータの自動更新」をクリックします。

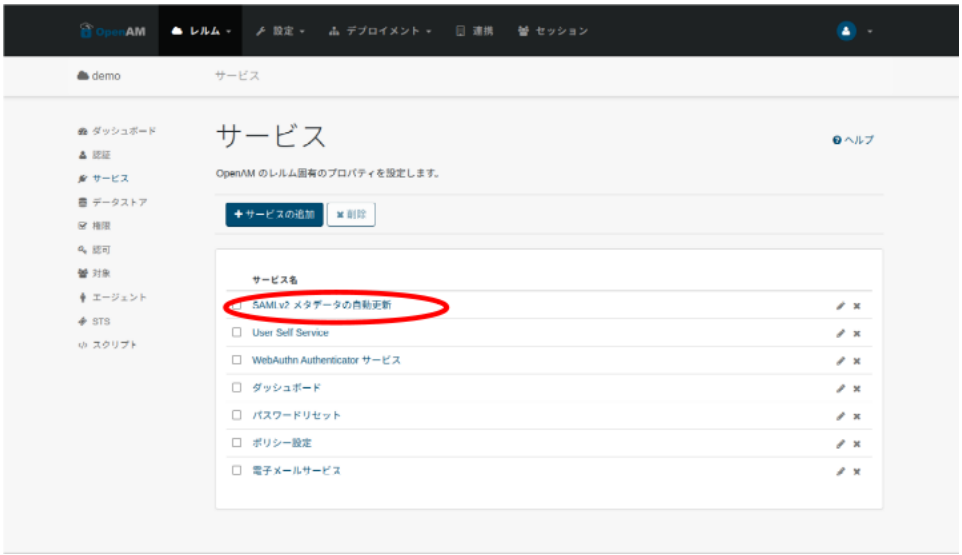

図 39　SAMLv2 メタデータの自動更新

- 「対象とするエンティティ」から対象 SP の EntityID を選択し、Del キーを押して削除します。

図 40　SAMLv2 メタデータの自動更新設定

- 「対象とするエンティティ」のリストから SP が削除されていることを確認し、「変更の保存」を押します。

図 41　確認して保存

- 「変更を保存しました」と表示されれば完了です。

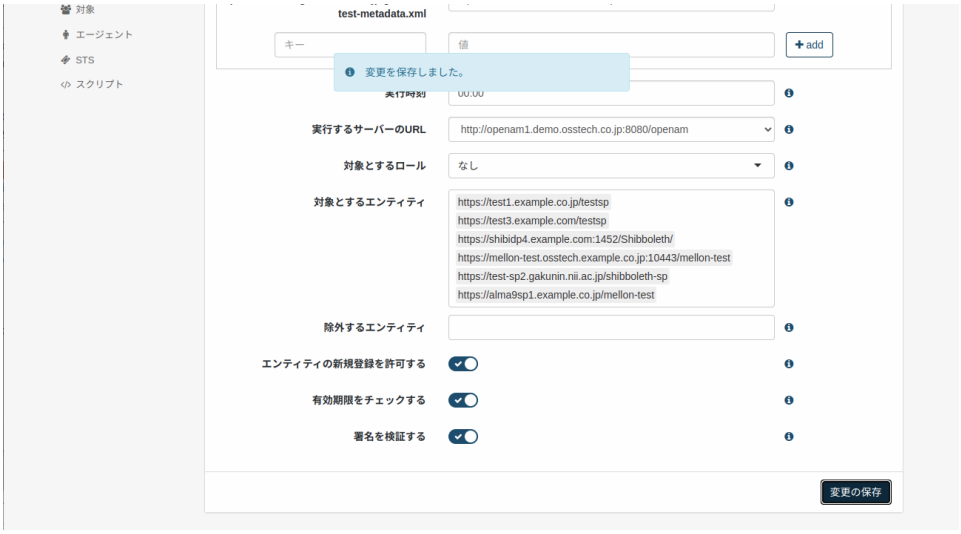

図 42　保存の完了

3.2.3 SP の設定の削除

利用しない SP の設定を削除します。

- OpenAM に管理者ユーザーでログインします。

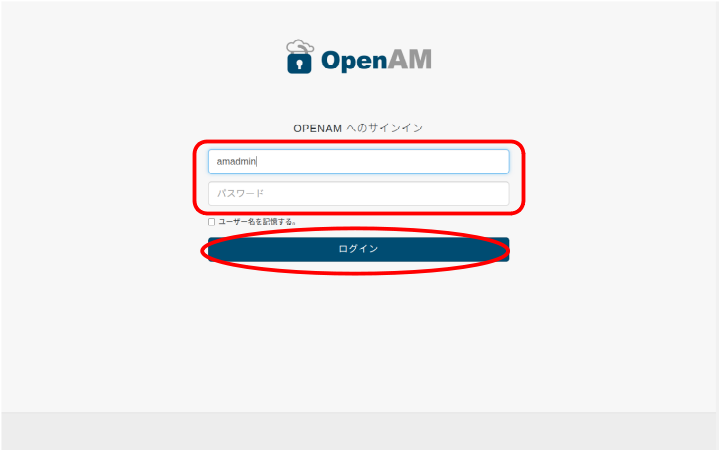

図 43　管理ユーザーログイン画面

- 画面上部のメニューから「連携」を押します。

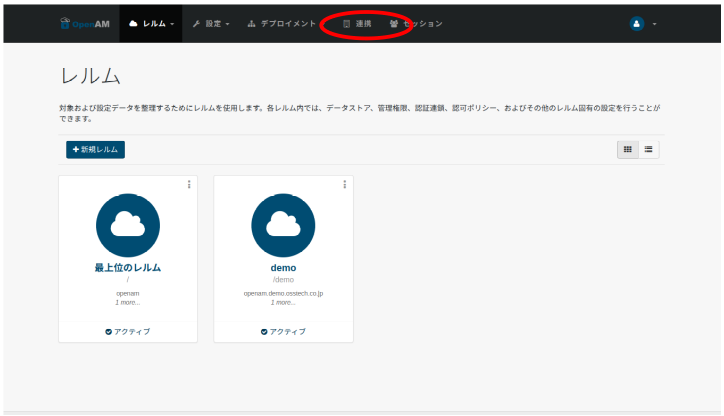

図 44　連携を押す

- エンティティプロバイダから「削除したい SP の EntityID(本書では https://test-sp1.gakunin.nii.ac.jp/shibboleth-sp)」をチェックし、「削除」を押します。

図 45　SP の EntityID を削除

- “【SP の EntityID】が削除されました。” と表示されれば完了です。

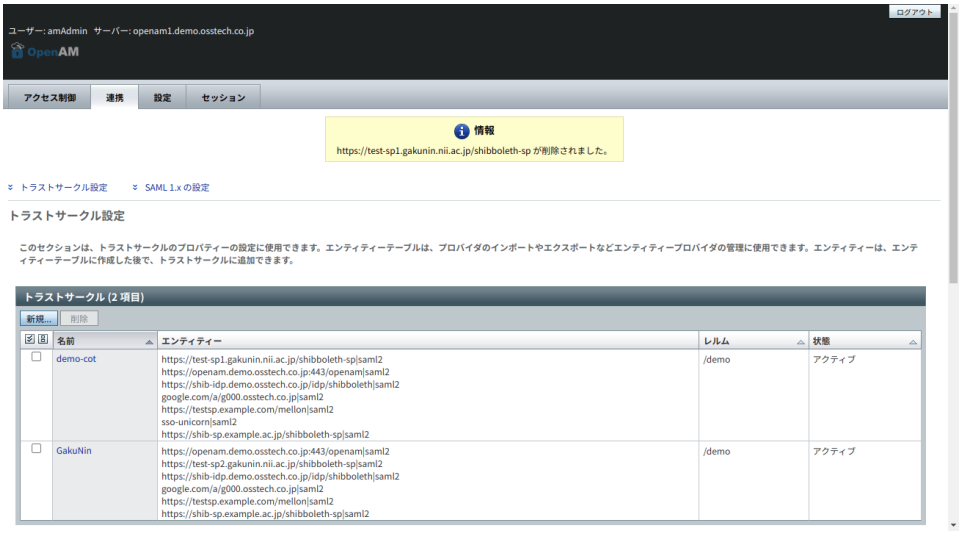

図 46　削除の完了

以上で、SP の削除は完了です。

3.3 IdP のサーバー証明書の更新

サーバー証明書の更新手順について説明します。作業の流れは学認の手順 (IdP Key Rollover) と 同様で、メタデータ伝播中に IdP が利用できない期間が発生しないようにします。手元に更新用のサーバー証明書および秘密鍵が準備されているものとします。

3.3.1 1 日目

3.3.1.1 Apache に対して証明書の更新

OpenAM サーバーの Apache の証明書を新しいものに差し替え、Apache の再起動を実施します。

3.3.1.2 学認申請システムにて証明書を追加

学認申請システム上で、予備の欄に更新用のサーバー証明書を登録します。すでに予備の欄に証明書が登録済みの場合、古いサーバー証明書が学認申請システム上に残っている状態です。今回のタイミングで整理し、学認申請システム上では、現在利用中のサーバー証明書と予備の欄での更新用のサーバー証明書を登録した状態としてください。

3.3.1.3 OpenAM にサーバー証明書の追加

更新用のサーバー証明書を OpenAM に追加します。更新用のサーバー証明書と秘密鍵を、署名鍵/暗号鍵を OpenAM のキーストアへインポートの手順を実施してキーストアにインポートします。

- 本書では、現在利用中の鍵ペアのエイリアス名を gakunin-cert-2023 、更新用のものを gakunin-cert-2024 とします。エイリアス名はご利用の環境に合わせて読み替えてください。
- OpenAM が冗長化構成の場合は、1 台の OpenAM でインポートを実施し、作業後に他のサーバーへキーストアファイルをコピーしてください。全ての OpenAM サーバーが同じキーストアファイルを利用する必要があります。

キーストアファイルの更新後、Tomcat の再起動を行います。

```
# systemctl restart osstech-tomcat
```

OpenAM のホスト IdP の設定を変更します。

- OpenAM に管理者ユーザーでログインします。

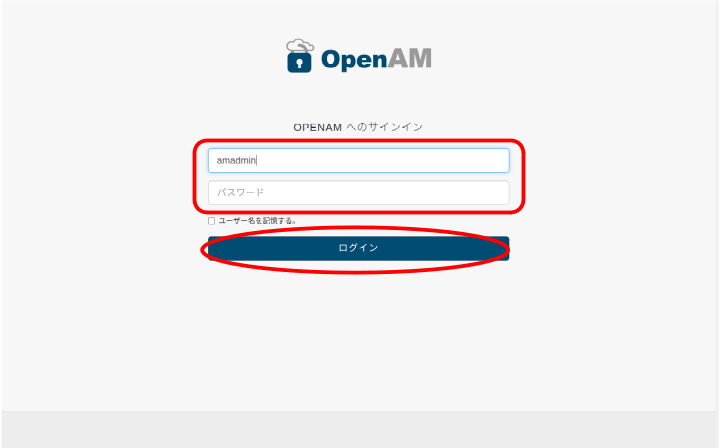

図 47　管理ユーザーログイン画面

- 画面上部のメニューから「連携」を押します。

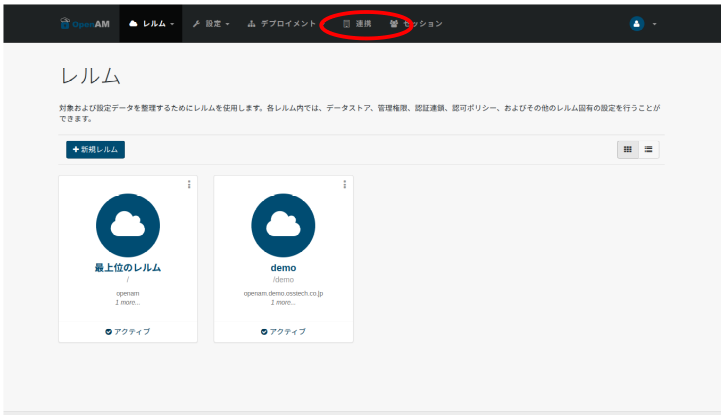

図 48　連携を押す

- エンティティプロバイダから、「ホスト IdP の EntityID(本書では https://shib-idp.demo.osstech.co.jp/idp/shibboleth)」をクリックします。

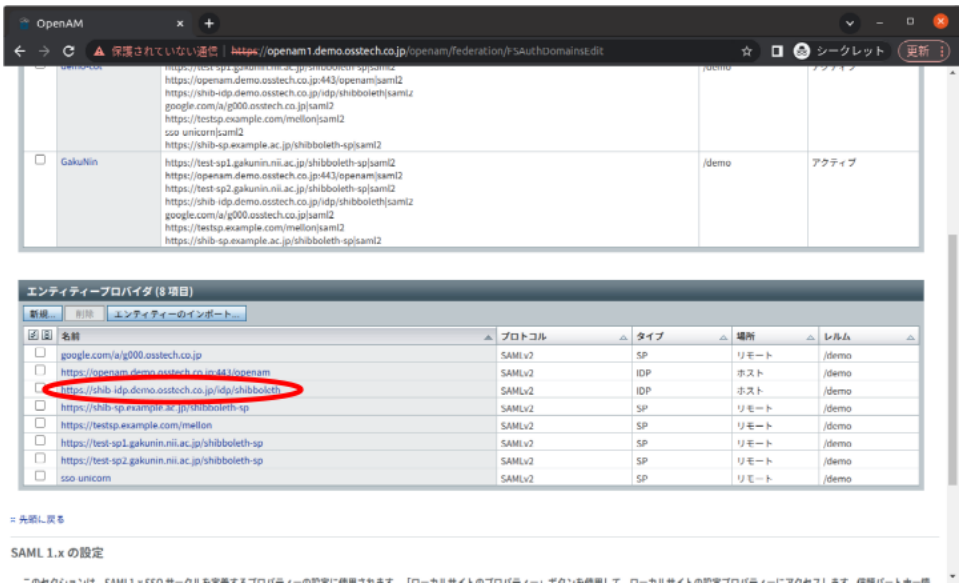

図 49　ホスト IdP の EntityID を選択

- 証明書エイリアスの署名の「新しい値」に更新用のエイリアス名を入力し、「追加」を押します。

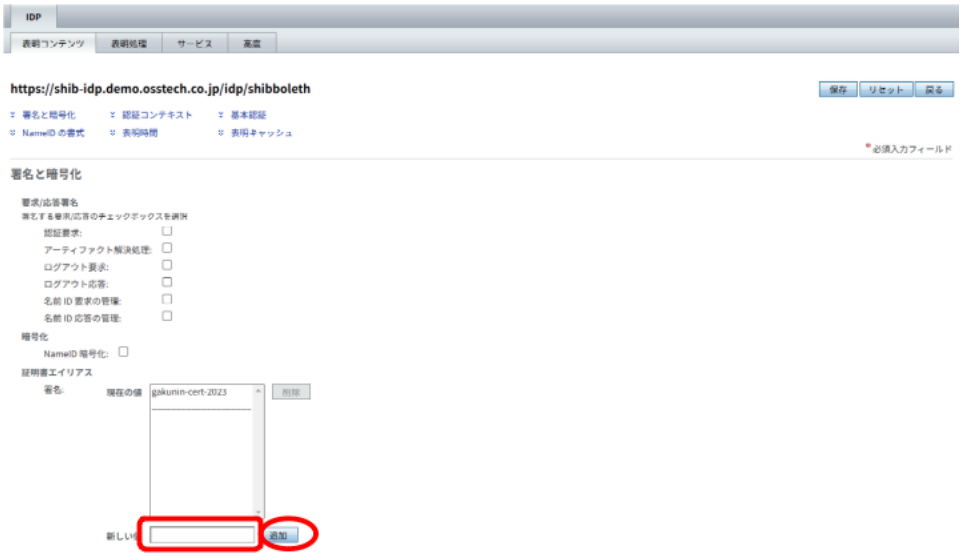

図 50　証明書の追加

- 証明書エイリアスの署名の現在の値のリストの 1 番目が「gakunin-cert-2023(現在利用中のエイリアス名)」2 番目が「gakunin-cert-2024(更新用のエイリアス名)」であることを確認し、「保存」を押します。

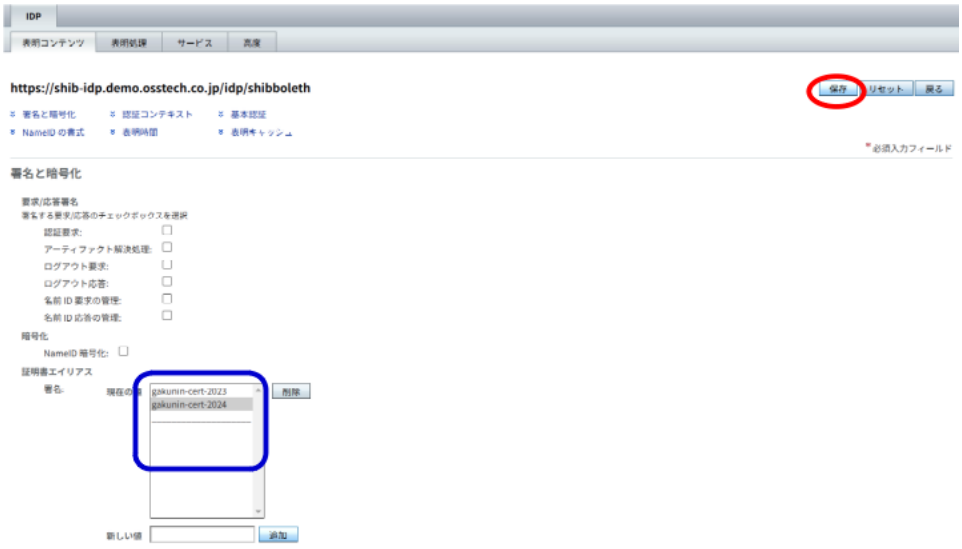

図 51　確認と保存

- “SAMLv2 アイデンティティープロバイダプロパティーが更新されました。” と表示されることを確認します。

図 52　更新の完了

- IdP のメタデータを表示します。メタデータの表示 URL は次のとおりです。[*8]
  - https://【FQDN】/【コンテキストパス】/saml2/jsp/exportmetadata.jsp?entityid=【IdP の EntityID】&realm=【レルム名】

表示されたメタデータにおいて、署名用 (<KeyDescriptor use="signing">) のサーバー証明書が新/旧 2 つあることを確認します。

図 53　OpenAM IdP メタデータ

作業は以上で完了です。設定を終えたら、任意の学認 SP が使えることを確認します。

[*8] 例として、OpenAM セットアップガイドに従って構築した OpenAM の場合、メタデータ表示 URL は次のとおりです。https://openam01.example.co.jp/openam/saml2/jsp/exportmetadata.jsp?entityid=https://shib-idp.demo.osstech.co.jp/idp/shibboleth&realm=/sso

3.3.2 X 日目

学認申請システムにて予備の欄に証明書を追加した申請が承認され、学認メタデータに反映された日を X 日目とします。

3.3.3 X + 15 日目

新しい証明書が含まれたメタデータが伝播されたため、OpenAM による SAML の署名として新しい証明書が利用されるよう設定します。

- OpenAM に管理者ユーザーでログインします。

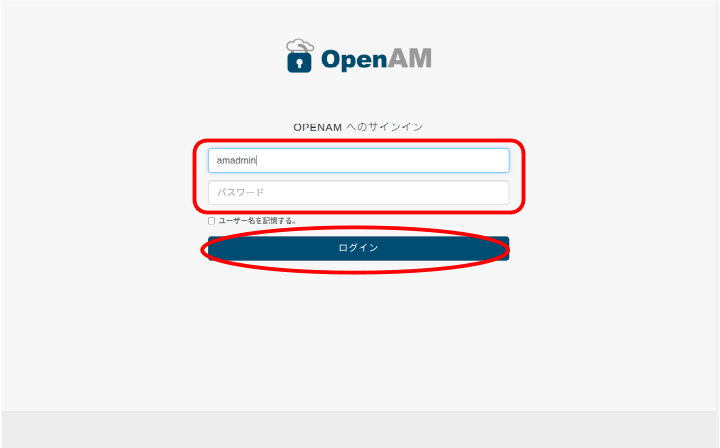

図 54　管理ユーザーログイン画面

- 画面上部のメニューから「連携」を押します。

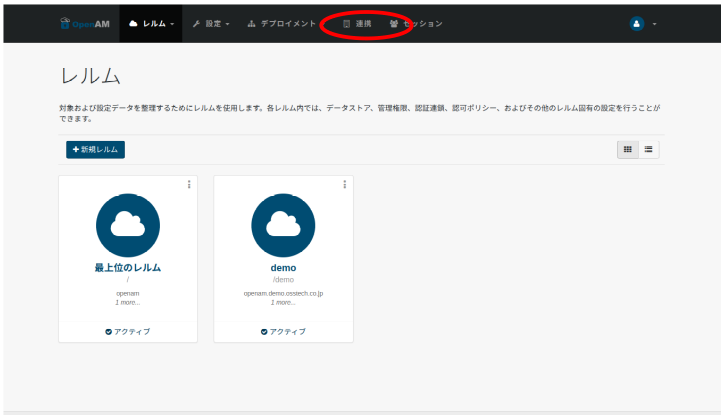

図 55　連携を押す

- エンティティプロバイダから「ホスト IdP の EntityID(本書では https://shib-idp.demo.osstech.co.jp/idp/shibboleth)」をクリックします。

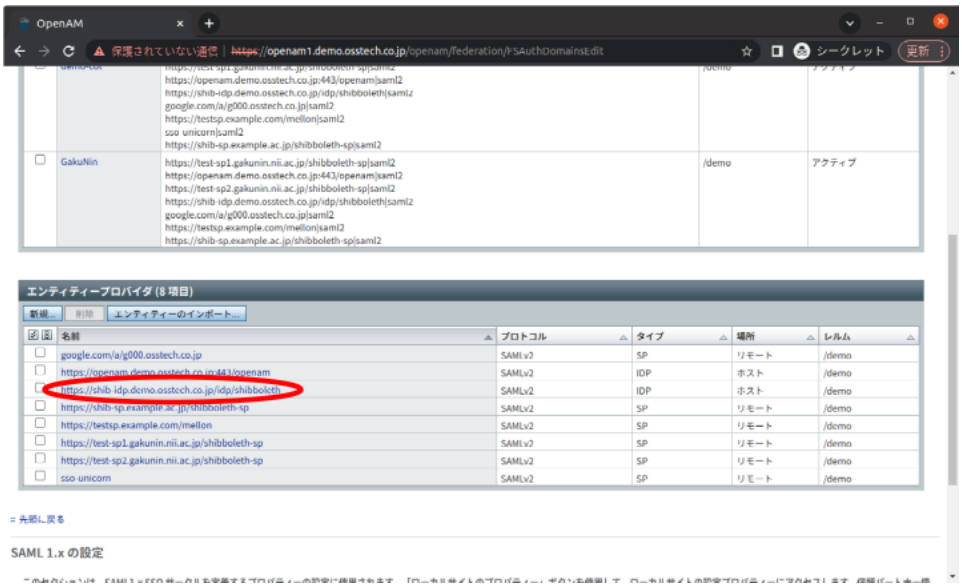

図 56　ホスト IdP の EntityID を選択

- 証明書エイリアスの署名の現在の値のリストの 1 番目の”gakunin-cert-2023(現在利用中のエイリアス名)“を選択し、「削除」を押します。

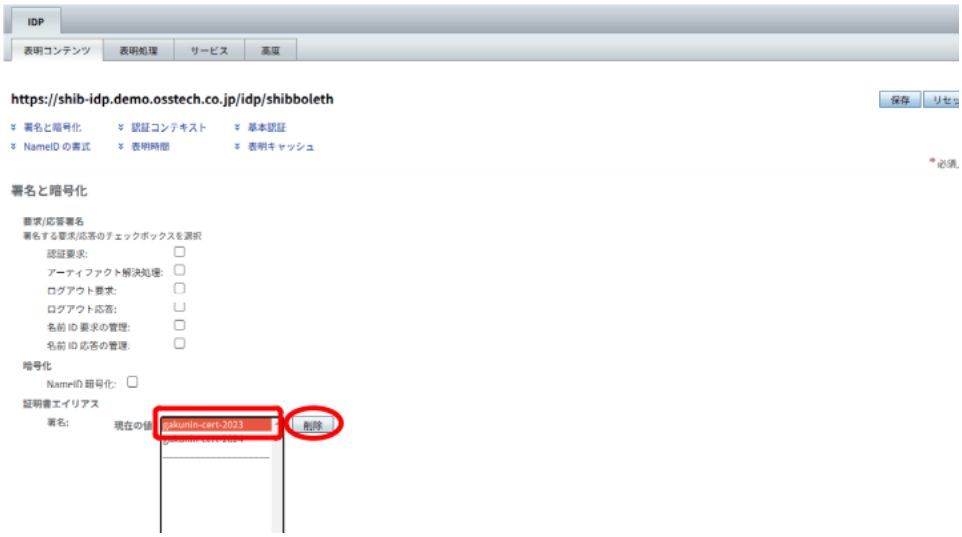

図 57　現在利用中のエイリアス名を削除

- 証明書エイリアスの署名の現在の値のリストが「gakunin-cert-2024(更新用のエイリアス名)」だけであることを確認し、「保存」を押します。

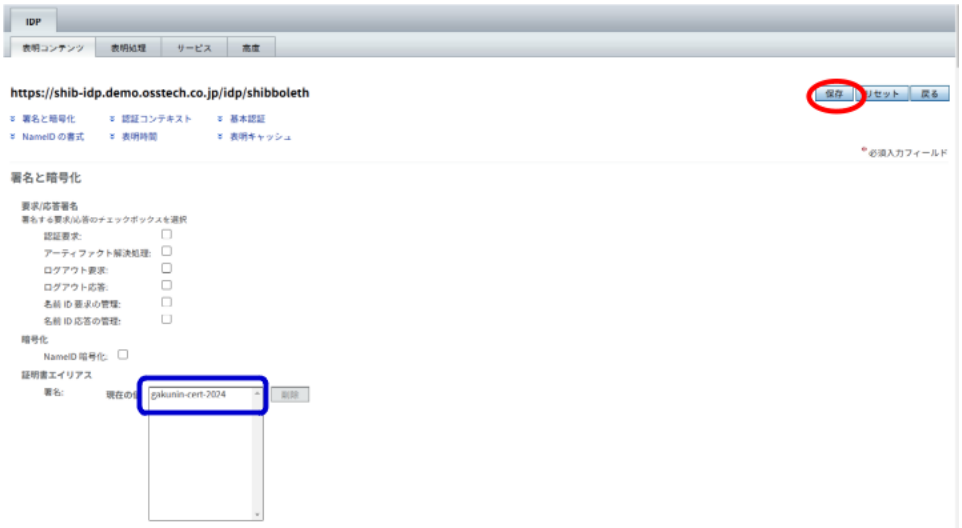

図 58　署名用エイリアスの更新

- “SAMLv2 アイデンティティープロバイダプロパティーが更新されました。” と表示されることを確認します。

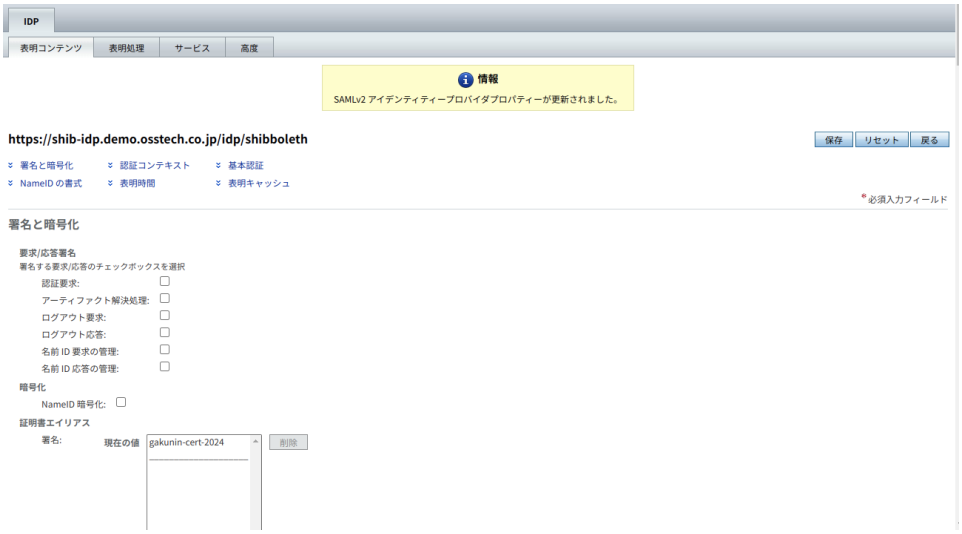

図 59　更新の完了

作業は以上で完了です。設定を終えたら、任意の学認 SP が使えることを確認します。問題がなければ、学認申請システムから古い証明書を削除し、新しい証明書を予備の欄から移

動してください。

4 改版履歴

- 2023 年 04 月 28 日 リビジョン 1.0
  - 初版作成