JCCH・セキュリティ・ソリューション・システムズ

プライベート認証局Gléas ホワイトペーパー

Microsoft Entra CBAによるシングルサインオン

(SAML連携)

Ver.1.0

2025 年 5 月

目次

1. はじめに

1.1. 本書について

本書では、弊社製品 プライベートCA Gléas で発行したクライアント証明書を利用して、Microsoft Entra ID の証明書ベース認証 (CBA)をおこないWebアプリケーションにアクセスする構成の設定例を記載します。

本書に記載の内容は、弊社の検証環境における動作を確認したものであり、あらゆる環境での動作を保証するものではありません。弊社製品を用いたシステム構築の一例としてご活用いただけますようお願いいたします。

弊社では試験用のクライアント証明書の提供も行っております。検証等で必要な場合は、最終項のお問い合わせ先までお気軽にご連絡ください。

1.2. 本書における環境

本書における手順は、以下の環境で動作確認を行っています。

- 認証基盤： Microsoft Entra ID

  ※以後「Entra ID」と記載します

- 認証局： JS3 プライベート認証局 Gléas（バージョン 2.7.1）

  ※以後「Gléas」と記載します

- Webサーバ： AlmaLinux9.4 / Apache 2.4.37

  (mod_ssl / mod_auth_mellon / mod_proxy / mod_headers)
  ※以後「Webサーバ」と記載します

- アプリケーションサーバ： AlmaLinux9.4 / Node.js v16.20.2

  ※以後「APサーバ」と記載します

- クライアント：Windows 10 Pro (22H2) / Microsoft Edge 136.0.3240.50

  ※以後「Windows」と記載します

- クライアント：iPhone 14 (iOS 18.3.1) / Safari 16.3

  ※以後「iPhone」と記載します

- クライアント：Google Pixel 7 Android15 / Chrome 135.0.7049.111

  ※以後「Android」と記載します

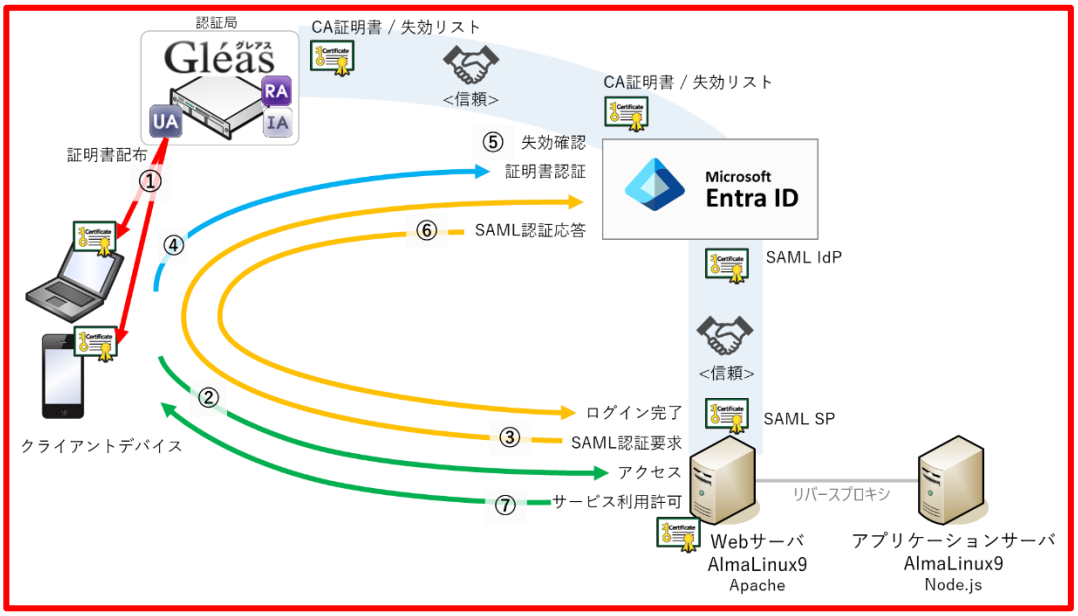
以下については、本書では説明を割愛します。

- Entra ID の基本設定
- Webサーバの基本設定（ネットワークや Apache の基本的な公開設定）
- APサーバの基本設定（ネットワークや Node.js の基本設定）
- Gléas のアカウント登録やクライアント証明書発行等の基本操作

これらについては、各製品のマニュアルをご参照いただくか、各製品を取り扱っている販売店にお問い合わせください。

1.3. 本書における構成

本書では、以下の構成で検証を行っています。



1. Gléas は、Web サーバ向けにサーバ証明書および SAML SP 証明書、クライアントデバイス向けにクライアント証明書を発行する。
2. Entra ID に、Gléas の CA 証明書を登録して証明書の発行元を信頼する。
3. Entra ID に、アプリケーションを登録する。
4. クライアントデバイスは、Gléas より証明書をインポートする。①
5. PC では Edge ブラウザ、iPhone では Safari ブラウザ、Android では Chrome ブラウザより Web サーバに HTTPS アクセスする。②
6. Web サーバは、Entra ID に SAML 認証要求。③
7. Entra ID は、クライアントデバイスに認証を要求しクライアント証明書認証を行う。④、⑤ 証明書を提示しない、期限切れ、または失効している、端末はクライアント証明書認証に失敗。
8. 認証成功すると、Entra ID は SAML 認証応答を発行、Web サーバは SAML 認証応答を受け取り、ログイン完了。⑥
9. Web サーバは、アプリケーションにリバースプロキシしてサービス利用可能となる。⑦

2. Web サーバの設定

SAML SPとして動作するWebサーバを設定します。

※本手順では、サーバで事前にApacheがインストールされていることが前提です。
※Apacheは mod_ssl, mod_auth_mellon, mod_proxy, mod_headers モジュールを有効化します。

2.1. アプリケーション構成

アプリケーションを公開するための構成を決定します。

アプリケーション名	SAMLデモアプリ
アプリケーションのURLパス	/saml_demo_app
サーバFQDN	sp.jcch-sss.com
プロキシURL	http://app-server:3000/saml_demo_app/
サーバ証明書のパス	/etc/httpd/conf/server.crt
サーバ秘密鍵のパス	/etc/httpd/conf/server.key
中間証明書のパス	/etc/httpd/conf/server-chain.crt
SAML SP証明書のパス	/etc/httpd/conf/saml_sp.crt
SAML SP秘密鍵のパス	/etc/httpd/conf/saml_sp.key
SAML SP の EntityID	https://sp.jcch-sss.com/saml_demo_app
SAML 名前ID形式	urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
Assertion Consumer ServiceのURLパス	/saml/saml_demo_app/postResponse
Single Logout ServiceのURLパス	/saml/saml_demo_app/logout
SAML SPメタデータのパス	/etc/httpd/conf/sp-metadata.xml
SAML IdPメタデータのパス	/etc/httpd/conf/idp-metadata.xml
アクセスログのパス	/etc/httpd/logs/saml_demo_app-access_log
エラーログのパス	/etc/httpd/logs/saml_demo_app-error_log
アプリケーションのサインオンURL	https://sp.jcch-sss.com/saml_demo_app/
認証ユーザをあらわすリクエストヘッダ	x-auth-user: “ユーザ プリンシパル名”

2.2. サーバ証明書の登録

Web サーバに適用するサーバ証明書を発行します。

※本手順では、Gléasで事前にサーバアカウントを作成してあることが前提です。
※アカウント名、ホスト名は、2.1 項の［サーバFQDN］を入力

Gléas RA（登録局）にログインし、該当のサーバアカウントのページへ移動します。

小メニューの[証明書発行]をクリックし、アカウントに対し証明書を発行します。

証明書詳細画面から[ダウンロード]をクリックし証明書をダウンロードします。

※ダウンロード時に証明書、秘密鍵を取り出す際のパスフレーズを指定します。

Gléasからダウンロードしたサーバ証明書 (.p12ファイル) をWebサーバにアップロードします。

.p12ファイルからPEM形式の証明書を取り出して配置します。

※OpenSSLで行なう例（パスフレーズの入力が必要となります)

```
openssl pkcs12 -in [.p12 ファイル] -nokeys -clcerts | openssl x509 -out /etc/httpd/conf/server.crt
chmod 644 /etc/httpd/conf/server.crt
```

.p12ファイルからPEM形式の秘密鍵を取り出して配置します。

※OpenSSLで行なう例（パスフレーズの入力が必要となります)

```
openssl pkcs12 -in [.p12 ファイル] -nodes -nocerts | openssl rsa -out /etc/httpd/conf/server.key
chmod 400 /etc/httpd/conf/server.key
```

2.3. SAML SP 証明書の登録

SAML SPとして使用する署名用証明書をGléasから発行します。

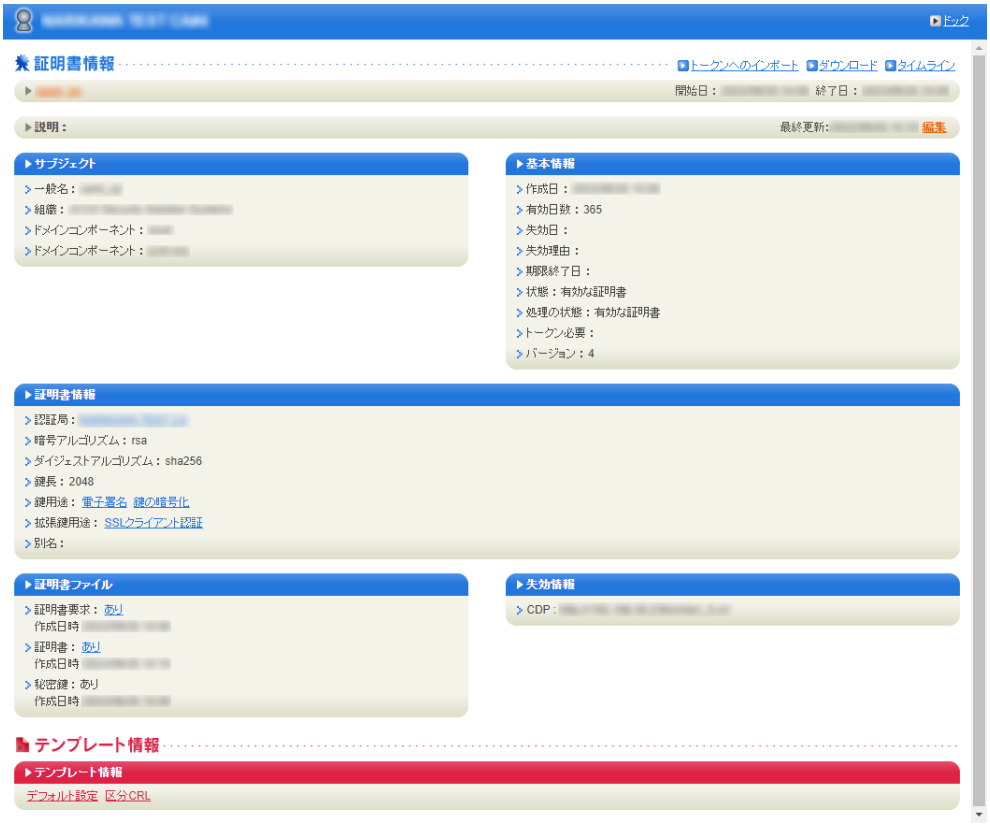
Gléas RA（登録局）にログインします。

[アカウント]>[アカウント新規作成]からアカウントを作成します。

※アカウント名は、2.1 項の［アプリケーションのURLパス］に準じたものを入力

小メニューの[証明書発行]をクリックし、アカウントに対し証明書を発行します。

証明書詳細画面から[ダウンロード]をクリックし証明書をダウンロードします。



※ダウンロード時に証明書、秘密鍵を取り出す際のパスフレーズを指定します。

GléasからダウンロードしたSP証明書（.p12ファイル）をWebサーバにアップロードします。

.p12ファイルからPEM形式の証明書を取り出して配置します。

※OpenSSLで行なう例（パスフレーズの入力が必要となります）

```
openssl pkcs12 -in [p12 ファイル] -nokeys -clcerts | openssl x509 -out /etc/httpd/conf/saml_sp.crt
chmod 644 /etc/httpd/conf/saml_sp.crt
```

.p12ファイルからPEM形式の秘密鍵を取り出して配置します。

※OpenSSLで行なう例（パスフレーズの入力が必要となります）

```
openssl pkcs12 -in [p12 ファイル] -nodes -nocerts | openssl rsa -out /etc/httpd/conf/saml_sp.key
chmod 400 /etc/httpd/conf/saml_sp.key
```

2.4. SAML SP メタデータの登録

SAML SP を定義するためのメタデータ（.xmlファイル）を作成します。

※2.1 項の構成に準じた SAML SP メタデータを作成するコマンド実行例

```
OUTFILE=/etc/httpd/conf/sp-metadata.xml
APP=saml_demo_app
FQDN=sp.jcch-sss.com
ENTITYID=https://sp.jcch-sss.com/$APP
SP_ENDPOINT=/saml/$APP
SP_CERT=/etc/httpd/conf/saml_sp.crt
X509=`openssl x509 -in $SP_CERT | grep -v '^-----'`

cat << EOS > $OUTFILE
<EntityDescriptor entityID="$ENTITYID"
                  xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
                  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
                   AuthnRequestsSigned="true">
    <KeyDescriptor use="encryption">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>$X509</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </KeyDescriptor>
    <KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>$X509</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </KeyDescriptor>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
                         Location="https://$FQDN$SP_ENDPOINT/logout"/>
    <NameIDFormat>
      urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
    </NameIDFormat>
    <AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
                              Location="https://$FQDN$SP_ENDPOINT/postResponse"
                              index="0"/>
  </SPSSODescriptor>
</EntityDescriptor>
EOS

chmod 644 $OUTFILE
```

作成された SAML SP メタデータダウンロードします。

※上記の例では /etc/httpd/conf/sp-metadata.xml

2.5. ヴァーチャルホスト設定

アプリケーションを公開するためのヴァーチャルホストを設定します。

※2.1 項の構成に準じたヴァーチャルホスト設定を作成するコマンド実行例

```
OUTFILE=/etc/httpd/conf.d/vhost-saml_demo_app.conf
APP=saml_demo_app
FQDN=sp.jcch-sss.com
SERVER_CERT=/etc/httpd/conf/server.crt
SERVER_PKEY=/etc/httpd/conf/server.key
SERVER_CHAIN=/etc/httpd/conf/server-chain.crt
IdP_METADATA=/etc/httpd/conf/idp-metadata.xml
SP_METADATA=/etc/httpd/conf/sp-metadata.xml
SP_CERT=/etc/httpd/conf/saml_sp.crt
SP_PKEY=/etc/httpd/conf/saml_sp.key
SP_ENDPOINT=/saml/$APP
PROXYBASE=http://app-server:3000

cat << EOS > $OUTFILE
<VirtualHost *:443>
  ServerName $FQDN
  CustomLog logs/$APP-access_log combined
  ErrorLog   logs/$APP-error_log

  SSLEngine on
  SSLCertificateFile          $SERVER_CERT
  SSLCertificateKeyFile       $SERVER_PKEY
  SSLCertificateChainFile     $SERVER_CHAIN

  <Location />
      MellonEnable                info
      MellonEndpointPath          $SP_ENDPOINT
      MellonSPPrivateKeyFile      $SP_PKEY
      MellonSPCertFile            $SP_CERT
      MellonSPMetadataFile        $SP_METADATA
      MellonIdPMetadataFile       $IdP_METADATA
  </Location>
  <Location /$APP>
      AuthType        Mellon
      Require         valid-user
      MellonEnable    auth
      RequestHeader set X-AUTH-USER %{MELLON_NAME_ID}e env=MELLON_NAME_ID
      RequestHeader set X-Forwarded-Proto https
      RequestHeader set X-Forwarded-Port 443
  </Location>

  ProxyPreserveHost On
  ProxyPass             /$APP $PROXYBASE/$APP
  ProxyPassReverse      /$APP $PROXYBASE/$APP
</VirtualHost>
EOS

chmod 644 $OUTFILE
```

3. Entra ID の設定

3.1. 認証局を登録

証明書ベース認証と連携する認証局を登録します。

本手順の前にGléasよりルート証明書をダウンロードします。

※GléasのデフォルトCAのルート証明書（PEM形式）のダウンロードURLは以下となります
http://[GléasのFQDN]/crl/ia1.pem

Microsoft Entra 管理センター にログインします。

メニュー [保護] > [セキュリティセンター] を選択します。

[認証局] を選択し、[アップロード] をクリックします。

- [証明書] に Gléas のルート証明書を指定

  ※拡張子が .cer でないとアップロードできないので、.pem を .cer に変更して指定
- [ルートCA証明書である] に [はい] を選択
- [証明書失効リストのURL] にCRL配布点のURLを入力

  ※GléasのデフォルトCRL配布点のURLは以下となります
  http://[GléasのFQDN]/crl/ia1.crl
- [デルタ証明書失効リストの URL] は指定しない
- [追加] をクリック

認証局が登録されました。

3.2. グループの作成

証明書ベース認証を行う対象となるセキュリティグループを作成します。

Microsoft Entra 管理センター にログインします。

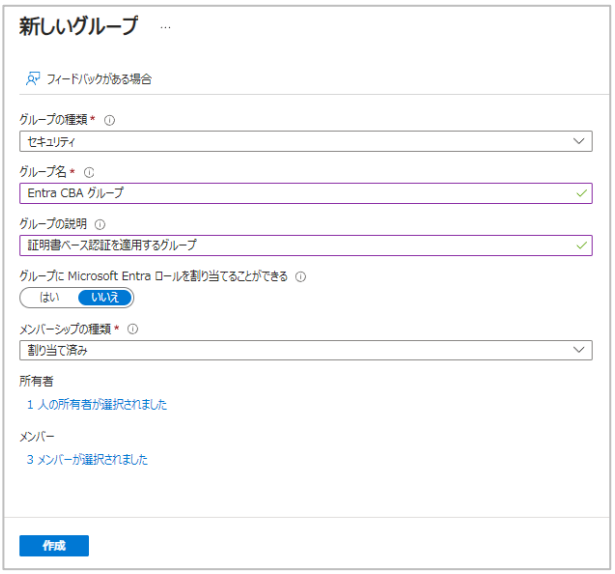
メニュー [グループ] > [すべてのグループ]を選択します。

[新しいグループ] をクリックします。

- [グループの種類] に [セキュリティ] を選択
- [グループの名] に任意の名前を入力

  ※例) "Entra CBA グループ"
- [グループの説明] に任意の説明を入力

  ※例) "証明書ベース認証を適用するグループ"
- [メンバーシップの種類] に [割り当て済み] を選択
- [所有者] の [所有者が選択されていません] をクリックして、セキュリティグループの所有者となるユーザーを選択
- [メンバー] の [メンバーが選択されていません] をクリックして、セキュリティグループの所属メンバーとなるユーザーを選択
- [作成] をクリック



セキュリティグループが作成されました。

3.3. 証明書ベース認証の有効化

作成したセキュリティグループに対して証明書ベース認証を有効化します。

Microsoft Entra 管理センター にログインします。

メニュー [保護] > [認証方法]を選択します。

[ポリシー] を選択し、[証明書ベースの認証] をクリックします。

[有効化およびターゲット] タブを選択します。

- [有効にする] を ON
- [含める] タブの [ターゲット] の [グループの選択] を選択
- [グループの追加] をクリックして作成したセキュリティグループを選択
- [保存] をクリック

3.3.1. 認証強度を構成

証明書ベース認証の認証強度を設定するポリシーを構成します。

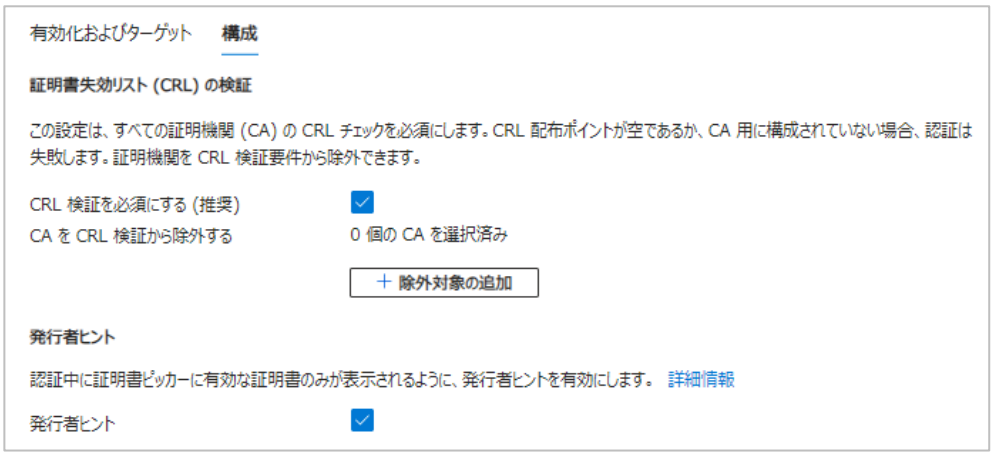
Microsoft Entra 管理センター にログインします。

メニュー [保護] > [認証方法]を選択します。

[ポリシー] を選択し、[証明書ベースの認証] をクリックします。

[構成] タブを選択します。

- [CRL検証を必須にする] をチェック
- [発行者ヒント] をチェック



- 認証バインド [保護レベル] に [多要素認証] を選択
- 認証バインド [必須のアフィニティバインド] に [低] を選択
- 認証バインド [＋規則の追加] をクリック

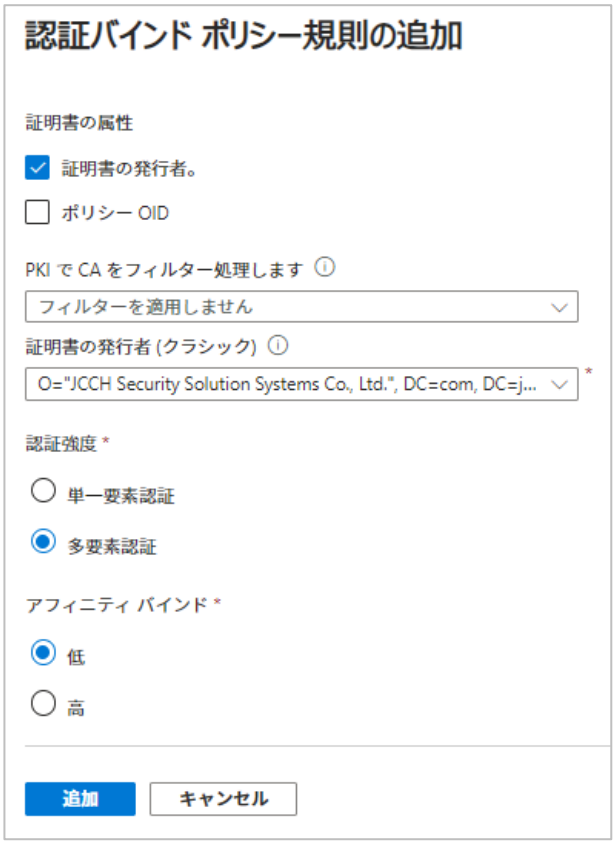
- 証明書の属性 で [証明書の発行者] を選択
- [証明書の発行者] に先に登録した証明機関を選択
- [認証強度] に [多要素認証] を選択

  ※本構成では証明書を用いた認証を多要素認証として扱うように設定しています。

- [アフィニティ バインド] に [低] を選択

  ※本構成では証明書の別名(UPN) と Entra ID ユーザーの userPrincipalName 属性でバインドするため、「低」 を指定しています。

- [追加] をクリック



- [保存] をクリック

3.3.2.証明書とユーザーの紐づけを構成

証明書ベース認証でクライアント証明書と Entra ID ユーザーを紐づけるポリシーを構成します。

Microsoft Entra 管理センター にログインします。

メニュー [保護] > [認証方法]を選択します。

[ポリシー] を選択し、[証明書ベースの認証] をクリックします。

[構成] タブを選択します。

- ユーザー名バインド [+規則の追加] をクリック
- [証明書フィールド] に [PrincipalName] を選択
- [ユーザー属性] に [userPrincipalname] を選択

※本構成では証明書の別名(UPN) をEntra ID ユーザーのUserPrincipalName 属性と突合して認証します。
※証明書フィールド、ユーザー属性のマッピングの他の組み合わせは、Microsoft社の情報をご参照ください。

- [追加] をクリック

- [保存] をクリック

3.4. 条件付きアクセスを構成

アプリケーションへのアクセスに証明書ベース認証を強制するための設定を行います。

※条件付アクセスの利用には Entra ID Premium P1 または P2 ライセンスが必要となります。

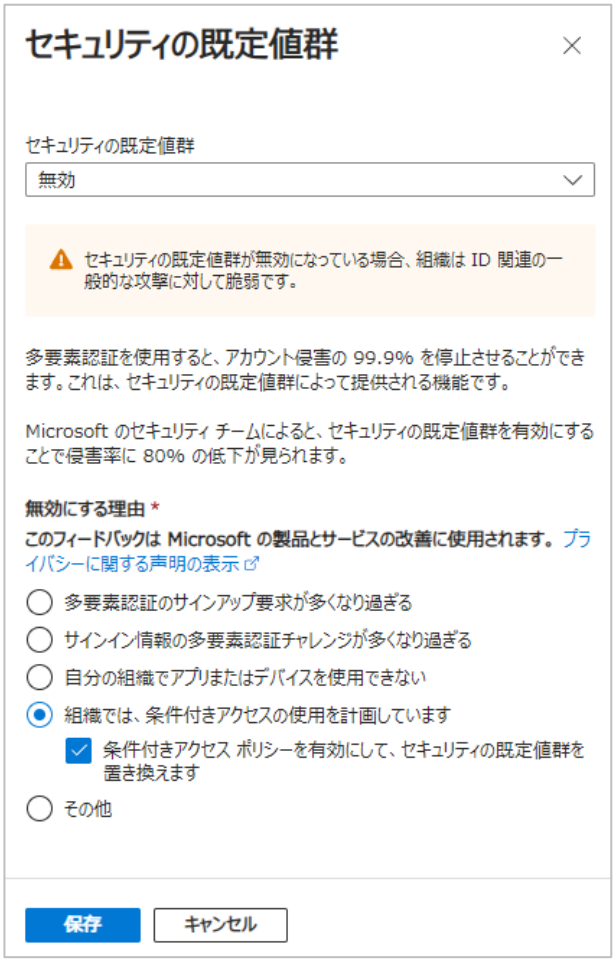
3.4.1. セキュリティの規定値群の無効化

「条件付きアクセス」機能を有効化するため、セキュリティの規定値群設定を無効化します。

Microsoft Entra 管理センター にログインします。

メニュー [ホーム] > [概要] を選択し、[プロパティ] タブを選択します。

- [セキュリティの規定値の管理] をクリック
- [セキュリティの規定値群] に [無効] を選択
- [無効にする理由] に [組織では、条件付きアクセスの使用を計画しています] を選択
- [条件付きアクセス ポリシーを有効にして、セキュリティの既定値群を置き換えます] をチェック
- [保存] をクリック
- 確認ダイアログで [無効化] をクリック


セキュリティの既定値群
セキュリティの既定値群
無効
セキュリティの既定値群が無効になっている場合、組織は ID 関連の一般的な攻撃に対して脆弱です。
多要素認証を使用すると、アカウント侵害の 99.9% を停止させることができます。これは、セキュリティの既定値群によって提供される機能です。
Microsoft のセキュリティ チームによると、セキュリティの既定値群を有効にすることで侵害率に 80% の低下が見られます。
無効にする理由 *
このフィードバックは Microsoft の製品とサービスの改善に使用されます。プライバシーに関する声明の表示
多要素認証のサインアップ要求が多くなり過ぎる
サインイン情報の多要素認証チャレンジが多くなり過ぎる
自分の組織でアプリまたはデバイスを使用できない
組織では、条件付きアクセスの使用を計画しています
条件付きアクセス ポリシーを有効にして、セキュリティの既定値群を置き換えます
その他
保存
キャンセル

3.4.2. 認証強度を構成

アプリケーションにアクセスする際に証明書提示を求めるように認証強度を構成します。

Microsoft Entra 管理センター にログインします。

メニュー [保護] > [条件付きアクセス] を選択します。

[認証強度] を選択し、[+新しい認証強度] をクリックします。

[構成] タブを選択します。

- [名前] に 任意の名前を入力

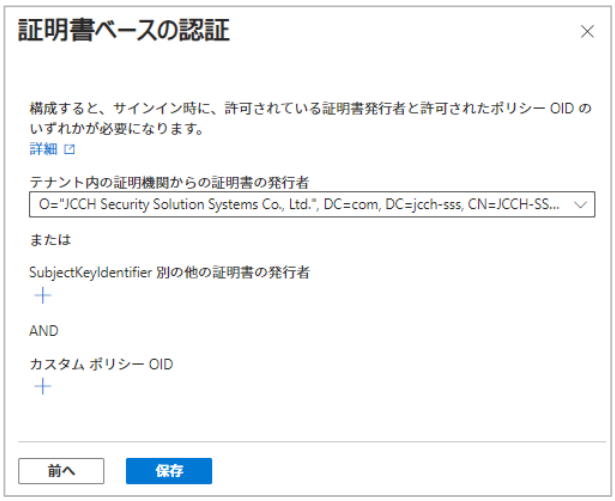
  ※例）"CBA必須"

- [説明] に 任意の説明を入力

  ※例）"証明書ベース認証を強制"

- [証明書ベースの認証（多要素）] をチェック

- 証明書ベース認証(多要素) の [詳細設定オプション] をクリック
- [テナント内の証明機関からの証明書の発行者]から3.1項で作成した認証局を選択
- [保存]をクリック



- [次へ] をクリックして [レビュー] タブに遷移
- [作成] をクリック

3.5. アプリケーションの登録

Entra ID にエンタープライズ アプリケーションを登録します。

Microsoft Entra 管理センター にログインします。

メニュー [アプリケーション] > [エンタープライズ アプリケーション]を選択します。

[すべてのアプリケーション] を選択し、[＋新しいアプリケーション] をクリックします。

[Microsoft Entra ギャラリーを参照する] で [+独自のアプリケーションの作成] をクリックします。

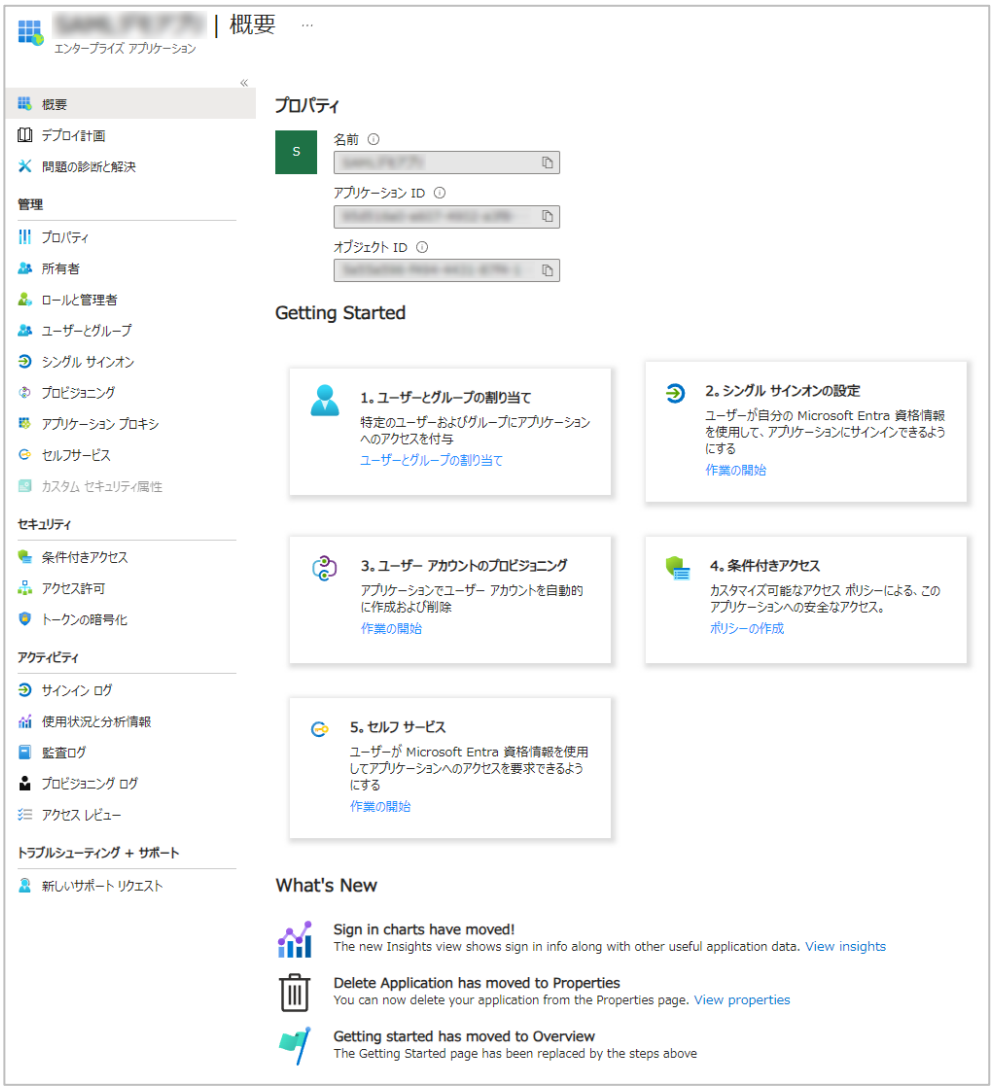
- [お使いのアプリの名前は何ですか?] に任意の名前を入力

  ※ここでは 2.1 項の [アプリケーション名] を入力

- [アプリケーションでどのような操作を行いたいですか?] に [ギャラリーに見つからないその他のアプリケーションを統合します] を選択

- [作成] をクリック

アプリケーションが登録されました。



3.5.1. アプリケーションに SSO を構成

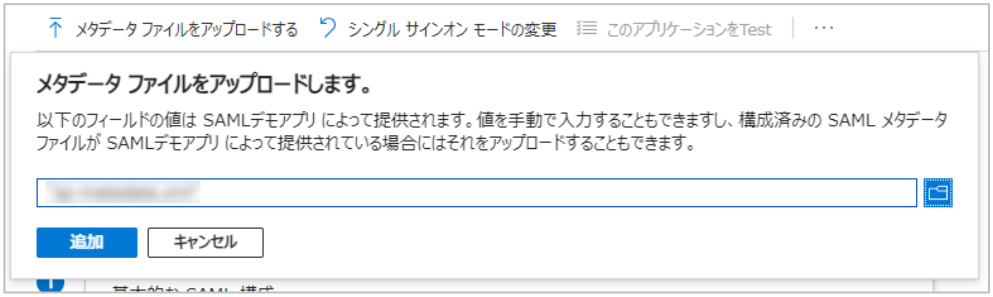
登録したアプリケーションにシングルサインオンを構成します。

シングルサインオンの設定 [作業の開始] をクリックします。

シングル サインオン方式の選択 画面で [SAML] を選択し、「SAML によるシングルサインオンのセットアップ」画面に遷移します。

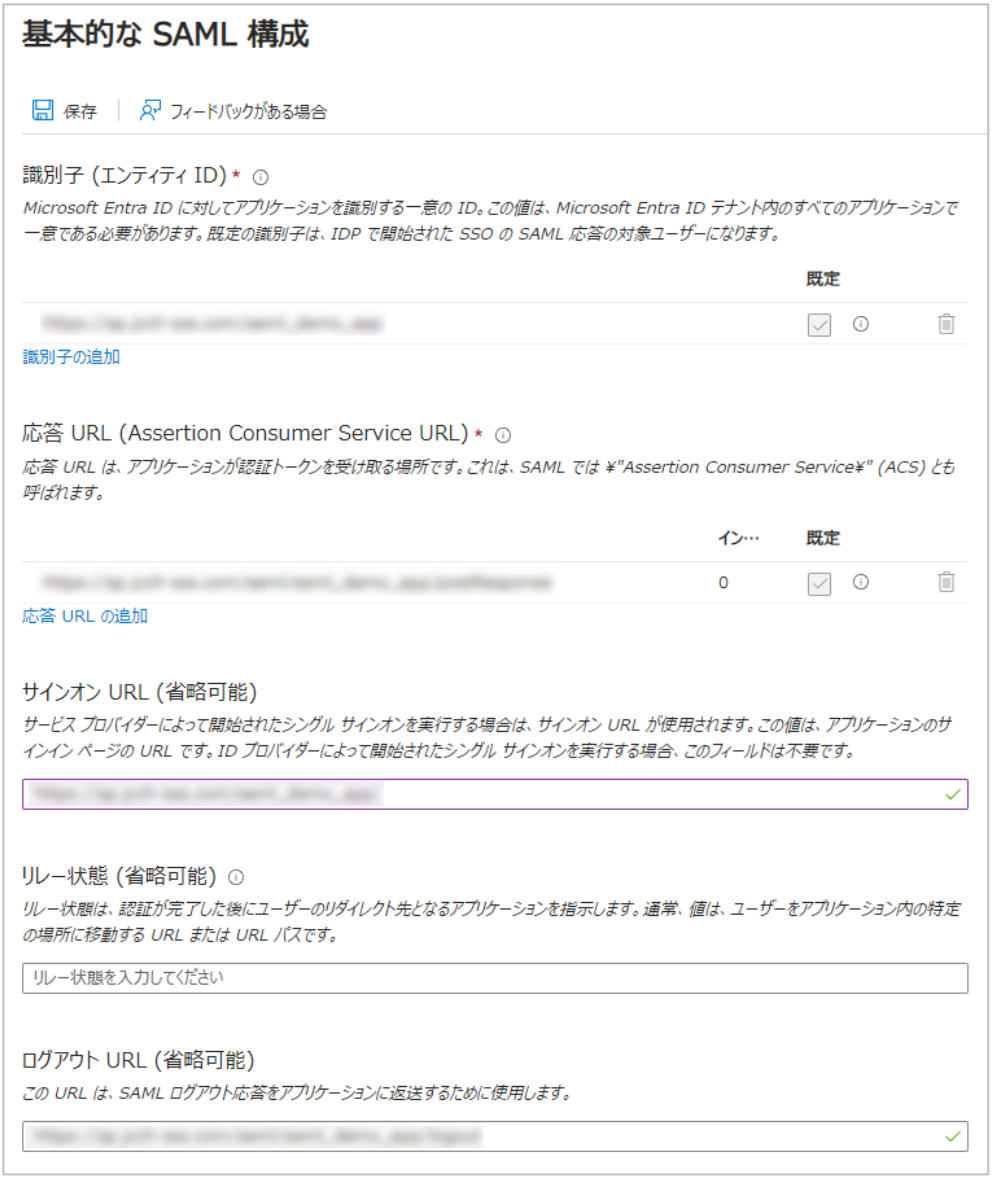
[メタデータファイルをアップロードする] をクリックします。

- 2.4 項でダウンロードした SAML SP メタデータを選択
- [追加] をクリック



- [識別子（エンティティ ID）] が入力されていることを確認
- [応答 URL (Assertion Consumer Service URL)] が入力されていることを確認
- [サインオン URL (省略可能)] にアプリケーションのサインオン URL を入力

  ※2.1 項の [アプリケーションのサインオン URL] を入力
- [リレー状態 (省略可能)] は入力されていないことを確認
- [ログアウト URL (省略可能)] が入力されていることを確認
- [保存] をクリック



[X] をクリックして閉じます。

※エンタープライズ アプリケーションのシングルサインオンのテスト確認が表示されたら［いいえ、後で test します］をクリックします。

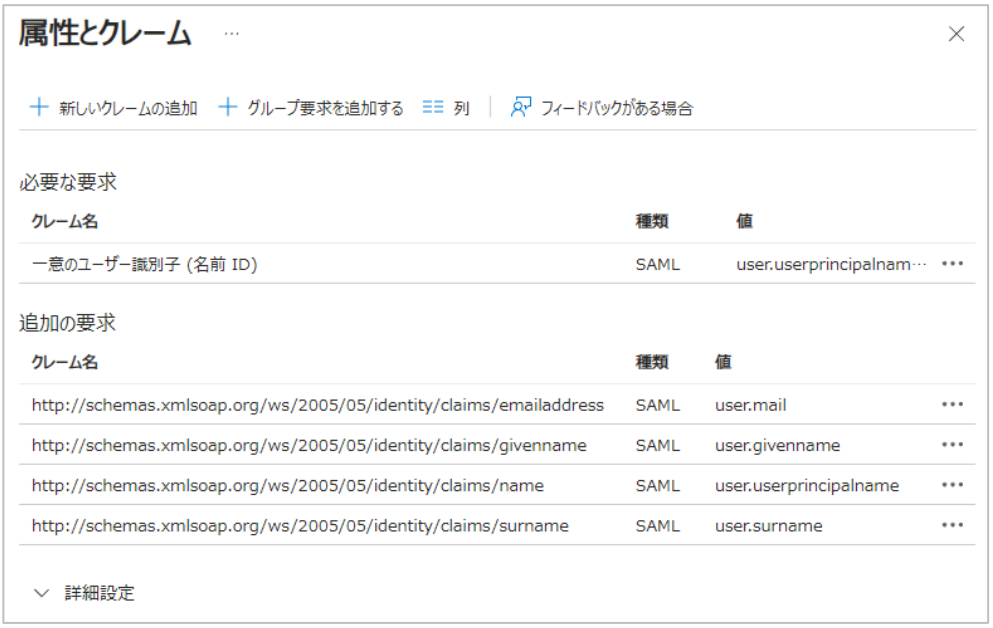
②属性とクレーム [編集]をクリックします。

- [必要な要求] の[一意のユーザー識別子(名前 ID)]の値をクリック
- [名前識別子の形式]に[電子メールアドレス]を選択
- [ソース] に [属性] を選択
- [ソース属性] に [user.userprincipalname] を選択
- [保存] をクリック

[X] をクリックして閉じます。

- [追加の要求] に以下を設定

名前	名前空間	ソース	ソース属性
name	http://schemas.xmlsoap.org/ws/2005/05/identity/claims	属性	user.userprincipalname
surname	http://schemas.xmlsoap.org/ws/2005/05/identity/claims	属性	user.surname
givenname	http://schemas.xmlsoap.org/ws/2005/05/identity/claims	属性	user.givenname
emailaddress	http://schemas.xmlsoap.org/ws/2005/05/identity/claims	属性	user.mail



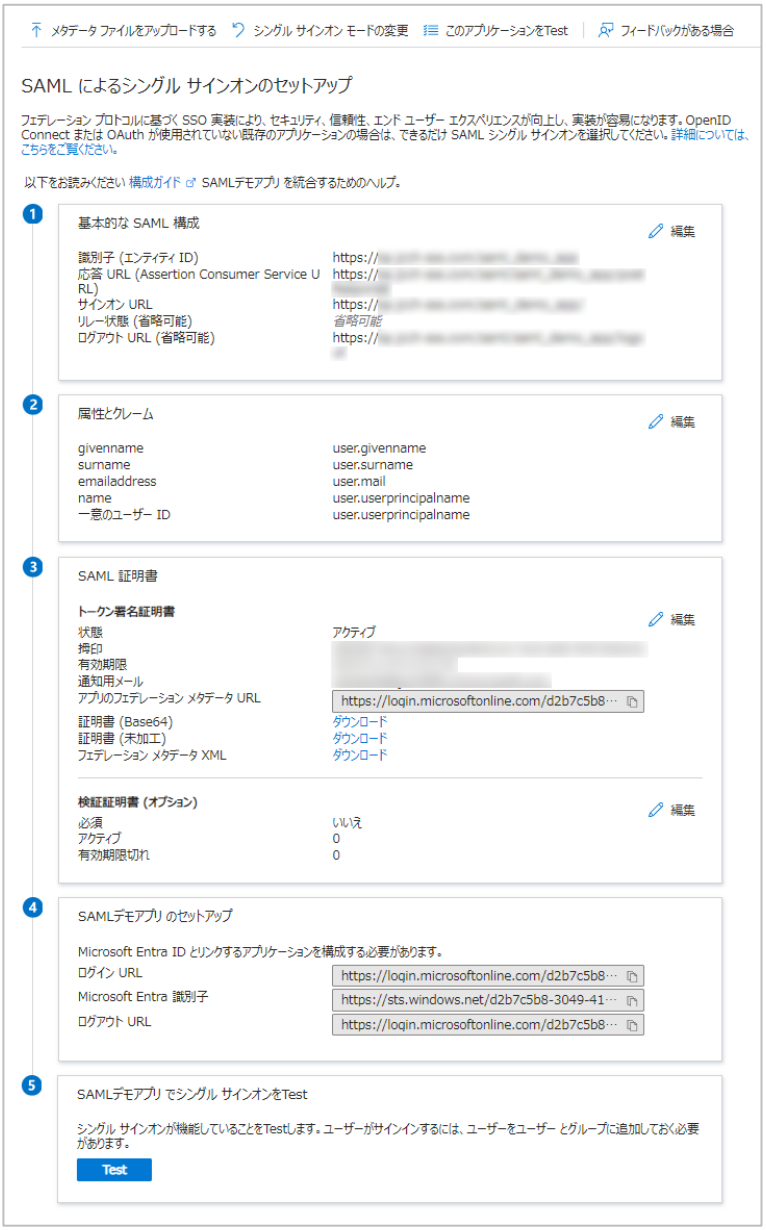
[X] をクリックして閉じます。

③SAML 証明書［トークン署名証明書]の[編集]をクリックします。

- [署名オプション] に [SAML 応答とアサーションへの署名]を選択
- [署名アルゴリズム] に [SHA-256] を選択
- [通知の電子メールアドレス] は未使用なのでそのまま
- [保存] をクリック

[X] をクリックして閉じます。

- [アプリのフェデレーション メタデータ URL] をコピーして保存しておく

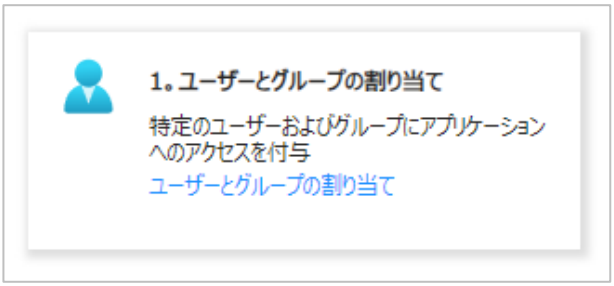


[X] をクリックして閉じます。

3.5.2.アプリケーションにユーザーを割り当て

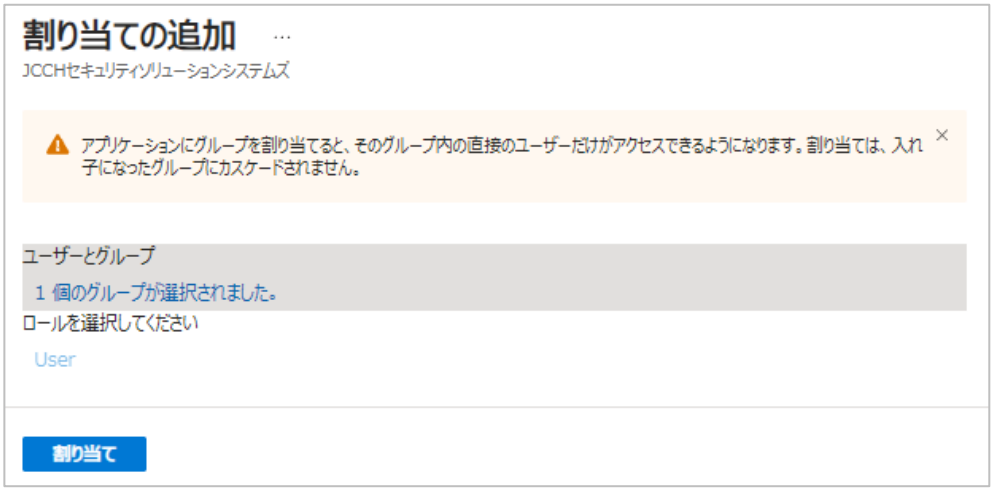
登録したアプリケーションにセキュリティグループを割り当て、ユーザーが利用できるようにします。

[ユーザーとグループの割り当て] をクリックします。



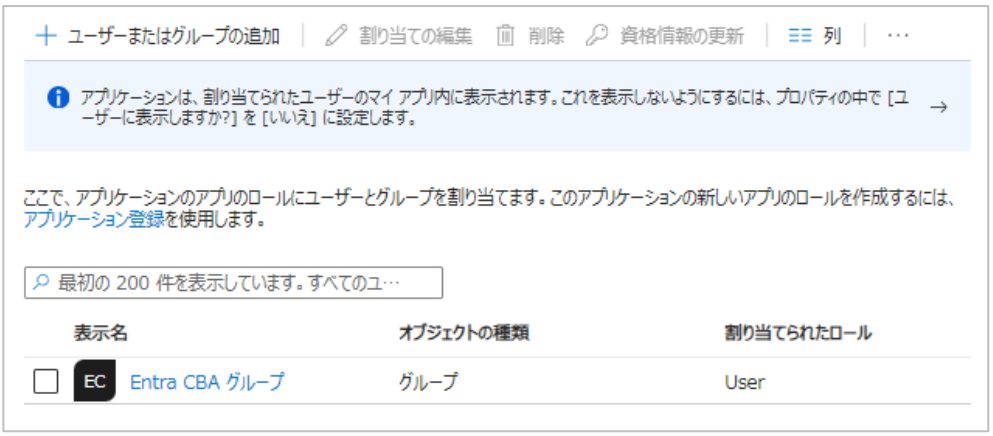
ユーザーとグループ 画面で [+ ユーザーまたはグループの追加] をクリックします。

- 3.2 項で作成したセキュリティグループを選択



- [割り当て]をクリック

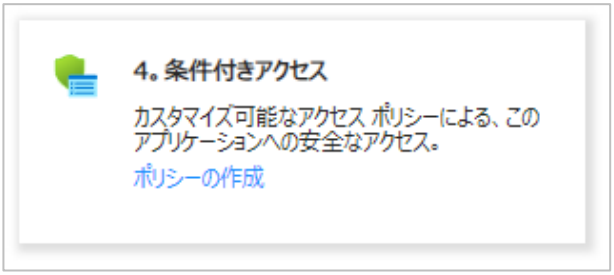
アプリケーションにセキュリティグループが割り当てられました。



3.5.3.アプリケーションへのアクセスポリシーを構成

登録したアプリケーションに条件付きアクセスを割当て、証明書ベース認証を強制します。

条件付きアクセス［ポリシーの作成］をクリックします。



条件付アクセス 画面で［+新しいポリシー］をクリックします。

- [名前] に 任意の名前を入力

  ※例）“CBA 強制ポリシー”

- 作成した認証強度を選択して[選択]をクリック

- 割り当て [ユーザー]で3.2項で作成したグループを選択

- 割り当て [ターゲットリソース]で対象アプリケーションが選択済みなことを確認

- 割り当て [ネットワーク]でアクセス元ネットワークを選択

  ※例）任意のネットワークまたは場所

- 割り当て [条件]の[デバイスプラットフォーム] でアクセス元デバイスを選択

  ※例）Windows、iOS、Android、macOS

- 割り当て [条件]の[クライアントアプリ] でアクセス元アプリを選択

  ※例）ブラウザー

- アクセス制御 [許可] で[認証強度が必要] をチェックして、3.4.2項で構成した認証強度を選択

- ポリシーの有効化 [オン] を選択

- [作成] をクリック


新規
条件付きアクセス ポリシー
シグナルを統合し、意思決定を行い、組織のポリシーを適用するために、条件付きアクセス ポリシーに基づいてアクセスを制御します。詳細情報
名前 *
CBA 強制ポリシー
割り当て
ユーザー
組み込まれた特定のユーザー
ターゲット リソース
1 個のリソースが含められました
ネットワーク 新規
任意のネットワークまたは場所
条件
3 個の条件が選択されました
アクセス制御
許可
1 個のコントロールが選択されました
セッション
0 個のコントロールが選択されました
ポリシーの有効化
レポート専用
オン
オフ
作成

アプリケーションにアクセスポリシーが構成されました。

4. Web サーバの設定

4.1. SAML IdP メタデータを Web サーバに登録

Webサーバで、Entra IDから SAML IdP メタデータをダウンロードします。

※以下のコマンドで 2.1 項の [SAML IdPメタデータのパス] に配置
※IdPメタデータURL は3.5.1 項で取得した [アプリのフェデレーション メタデータ URL] を使用

```
curl -o /etc/httpd/conf/idp-metadata.xml [IdP メタデータ URL]
chmod 644 /etc/httpd/conf/idp-metadata.xml
```

4.2. Web サーバ起動

準備ができたらWeb サーバを起動します。

これによりWebサーバは SAML SP として動作します。

※以下のコマンドで Webサーバを起動

```
sudo systemctl enable httpd
sudo systemctl start httpd
```

5. AP サーバの設定

APサーバを設定します。

※本手順は、APサーバに Node.js がインストールされていることが前提となります。

5.1. 構成

検証用アプリケーションの構成を決めます。

Listen tcpポート番号	3000
ログインユーザー表示	認証済みEntra ID のユーザー プリンシパル名を出力
ログアウトリンク表示	Single Logout Service へのリンクを出力 リンクを踏むとログアウト後、Rootパスへリダイレクト
その他	HTTPリクエストヘッダを出力

5.2. 実装

アプリケーションを Node.js で実装します。

※以下のコマンドで 5.1 項の構成に準じたアプリケーションを実装

```
cat << 'EOS' > /usr/local/src/saml_demo_app.js
"use strict";

const http = require("http");
const listen_port = 3000;
const slo_path = "/saml/saml_demo_app/logout";
const return_to = "https://sp.jcch-sss.com/";

const server = http.createServer((request, response) => {
      var headers = "";
      Object.keys(request.headers).forEach((key) => {
         headers = headers + `<li>${key}: ${request.headers[key]}</li>¥n`
      });
      response.writeHead(200, {"Content-Type": "text/html; charset=UTF-8",
                               "Cache-Control": "no-cache"});
      response.write(`¥
<html><body>¥n¥
   <h1>Welcome! </h1>¥n¥
   <h4>${request.headers["x-auth-user"]}</h4>¥n¥
   <a href='${slo_path}?ReturnTo=${return_to}'>Logout</a>¥n¥
   <h3>Request Header</h3>
   <ul>¥
      ${headers}¥
   </ul>¥n¥
</body></html>¥n¥
`);
      response.end();
});

server.listen(listen_port);

console.log(`The server has started and is listening on port : ${listen_port}`);
EOS

chmod 644 /usr/local/src/saml_demo_app.js
```

5.3. AP サーバ起動

```
node /usr/local/src/saml_demo_app.js
```

6. Gléas の設定

6.1. 証明書テンプレートの設定

Entra ID 証明書ベース認証の要件を満たすように Gléas のデフォルトテンプレートを以下のように設定します。

※下記設定は、Gléas納品時等に弊社で設定を既におこなっている場合があります

証明書の属性	データベースの項目
発行局	[発行局名]
暗号アルゴリズム	RSA暗号
鍵長	2048bit
ダイジェストアルゴリズム	SHA256
有効日数	1年
鍵用途	電子署名、鍵の暗号化
拡張鍵用途	SSLクライアント認証
別名（プリンシパル名）	アカウント（プリンシパル名）

6.2. アカウント作成と証明書発行

クライアント証明書の発行対象となる Gléas アカウントを作成し、証明書を発行します。

Gléas RA（登録局）にログインします。

[アカウント]>[アカウント新規作成]メニューから[上級者向け設定」をクリックします。

- [その他の設定]の[証明書を発行する]をチェック
- [▸種類]から[CSV ファイル一括]を選択
- [アップロードする]にローカルの CSV ファイルを選択

※CSV ファイルは以下の形式

列名	値
cn	アカウント名 ※証明書のサブジェクト一般名となります ※UA のログインユーザ ID となります
sn	名前（姓）
givenname	名前（名）
password	パスワード ※UA のログインパスワードとなります
upn	プリンシパル名 ※証明書の別名（UPN）となります ※Entra ID のユーザー プリンシパル名と一致させてください

- [作成]をクリック

- 内容を確認し[実行]をクリック

[アカウント]>[登録申請者一覧]メニューを選択します。

※しばらくするとアップロードしたユーザ情報がアカウント登録申請として登録されます

- [全て許可する」をクリック
- [実行]をクリック

CSV の内容が Gléas アカウントとしてインポートされます。

※しばらくするとアップロードしたアカウントに対してクライアント証明書が自動的に発行されます

6.3. 証明書の配布設定（Windows 向け）

GléasのUA（申込局）より発行済み証明書をPCにインポートできるよう設定します。

※下記設定は、Gléas納品時等に弊社で設定を既におこなっている場合があります

Gléas RA（登録局）にログインします。

画面上部より[認証局]をクリックし認証局一覧画面に移動し、設定を行うUA（申込局）

をクリックします。

※実際はデフォルト申込局ではなく、その他の申込局の設定を編集します

申込局詳細画面が開くので、基本設定で以下の設定を行います。

- [証明書ストアへのインポート]をチェック
- 証明書ストアの選択で、[ユーザストア]を選択
- 証明書のインポートを一度のみに制限する場合は、[インポートワンスを利用する]に

  チェック

各項目の入力が終わったら、 [保存]をクリックします。

また、認証デバイス設定の以下項目にチェックがないことを確認します。

- iPhone/iPad の設定の、[iPhone / iPad 用 UA を利用する]
- Android の設定の、[Android 用 UA を利用する]

6.4. 証明書の配布設定（iPhone 向け）

GléasのUA (申込局) より発行済み証明書を iOS にインポートできるよう設定します。

※下記設定は、Gléas納品時等に弊社で設定を既におこなっている場合があります

Gléas RA（登録局）にログインします。

画面上部より[認証局]をクリックし[認証局一覧]画面に移動し、設定を行うUA (申込局)

をクリックします。

※実際はデフォルト申込局ではなく、その他の申込局の設定を編集します

[申込局詳細]画面が開くので、[基本設定]部分で以下の設定を行います。

- [ダウンロードを許可]をチェック
- [ダウンロード可能時間(分)]の設定・[インポートワンスを利用する]にチェック

この設定を行うと、GléasのUAからインポートから指定した時間 (分) を経過した後は、

構成プロファイルのダウンロードが不可能になります（インポートロック機能）。これ

により複数台のデバイスへの構成プロファイルのインストールを制限することができま

す。

設定完了後、[保存]をクリックし保存します。

[認証デバイス情報]の[iPhone/iPadの設定]までスクロールし、[iPhone/iPad用UAを利用する]をチェックします。

構成プロファイルに必要となる情報の入力画面が展開されるので、以下設定を行います。

【画面レイアウト】

- [iPhone用レイアウトを利用する]をチェック
- [ログインパスワードで証明書を保護]をチェック

【iPhone構成プロファイル基本設定】

- [名前]、[識別子]に任意の文字を入力（必須項目）

各項目の入力が終わったら、 [保存]をクリックします。

6.5. 証明書の配布設定（Android 向け）

GléasのUA（申込局）より発行済み証明書を Android にインポートできるよう設定します。

※下記設定は、Gléas納品時等に弊社で設定を既におこなっている場合があります

Gléas RA（登録局）にログインします。

画面上部より[認証局]をクリックし[認証局一覧]画面に移動し、設定を行うUA (申込局)をクリックします。

※実際はデフォルト申込局ではなく、その他の申込局の設定を編集します

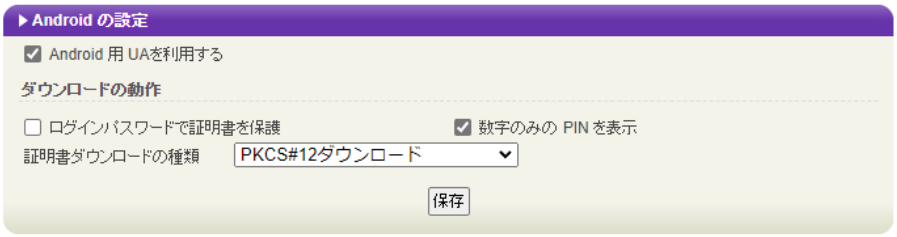
[申込局詳細]画面が開くので、[基本設定]部分で以下の設定を行います。

- [ダウンロードを許可]をチェック
- [ダウンロード可能時間(分)]の設定・[インポートワンスを利用する]にチェック

この設定を行うと、GléasのUAからインポートから指定した時間 (分) を経過した後は、証明書ファイルのダウンロードが不可能になります（インポートロック機能）。これにより複数台のデバイスへの証明書ファイルのインストールを制限することができます。

設定完了後、[保存]をクリックし保存します。

[認証デバイス情報]の[Androidの設定]までスクロールし、[Android用UAを利用する]をチェックします。



証明書のダウンロードに必要となる情報の入力画面が展開されるので、以下設定を行います。

- [数字のみのPINを表示]をチェック
- [証明書ダウンロードの種類]]を[PKCS#12ダウンロード]を選択

各項目の入力が終わったら、 [保存]をクリックします。

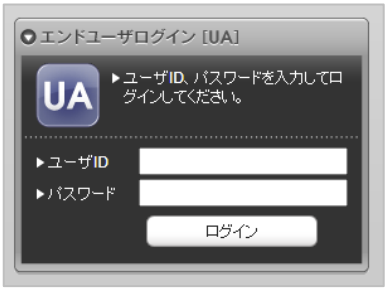
7. クライアントの設定

7.1. Windows にクライアント証明書をインポート

PCのブラウザ（Edge）で、UAにアクセスします。

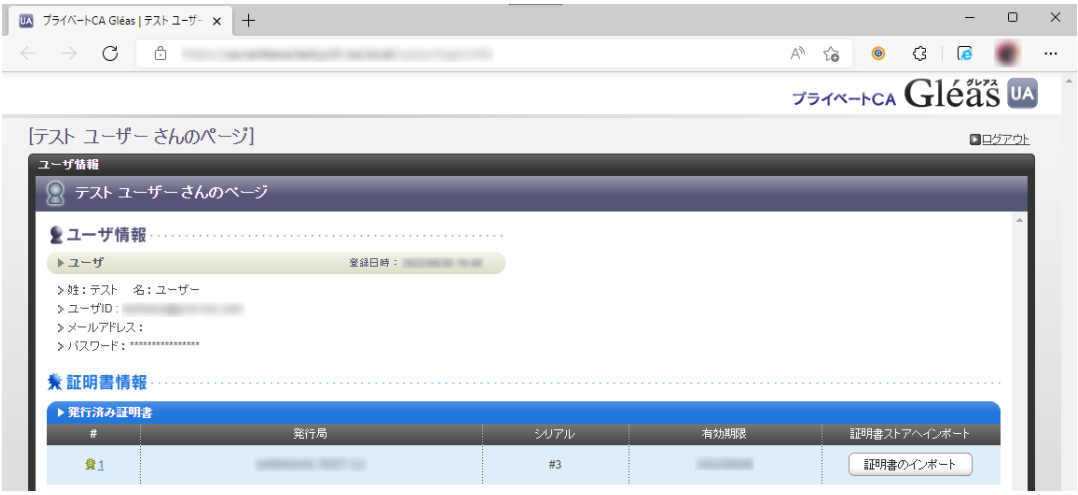
※URL https://[UAのFQDN]/[UAの名前]/ua

ログイン画面が表示されるので、ユーザIDとパスワードを入力しログインします。

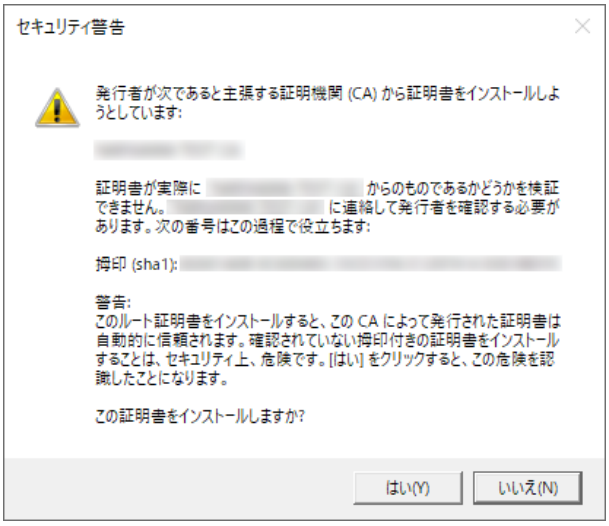


ログインすると、ユーザ専用ページが表示されます。

[証明書のインポート]ボタンをクリックすると、クライアント証明書のインポートが行われます。



※証明書インポート時にルート証明書のインポート警告が出現する場合は、システム管理者に拇印を確認するなど正当性を確認してから[はい]をクリックします



インポートワンス機能を有効にしている場合は、インポート完了後に強制的にログアウトさせられます。再ログインしても[証明書のインポート]ボタンは表示されず、再度ログインしてインポートを行うことはできません。

7.2. iPhone にクライアント証明書をインポート

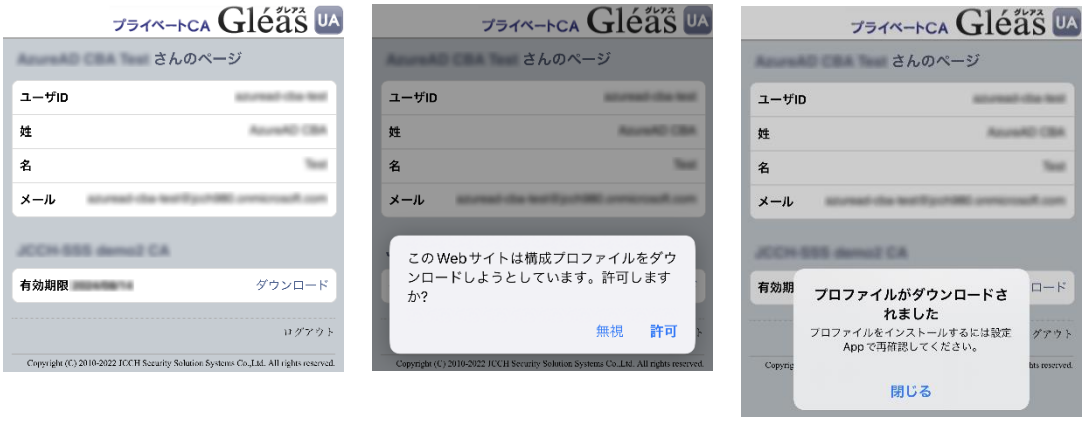
iPhoneのブラウザ（Safari）で、UAにアクセスします。

※URL https://[UAのFQDN]/[UAの名前]/ua

ログイン画面が表示されるので、ユーザIDとパスワードを入力しログインします。

ログインすると、ユーザ専用ページが表示されます。

[ダウンロード]をタップし、構成プロファイルのダウンロードをおこないます。



※インポートロックを有効にしている場合は、この時点からカウントが開始されます

画面の表示にしたがい設定を開くと、プロファイルがダウンロードされた旨が表示されるので、インストールをおこないます。

[インストール]をタップして続行してください。

インストール中にルート証明書のインストール確認画面が現れるので、内容を確認し[インストール]をタップして続行してください。

※ここでインストールされるルート証明書は、通常のケースではGléasのルート認証局証明書になります

インストール完了画面になりますので、[完了]をタップして終了します。

なお [詳細]をタップすると、インストールされた証明書情報を見ることができます。

必要に応じて確認してください。

Safariに戻り、[ログアウト]をタップしてUAからログアウトします。

以上で、iPhoneでの構成プロファイルのインストールは終了です。

なお、インポートロックを有効にしている場合、[ダウンロード]をタップした時点より管理者の指定した時間を経過した後にUAに再ログインすると、以下の通り「ダウンロード済み」という表記に変わり、以後のダウンロードは一切不可となります。

7.3. Android にクライアント証明書をインポート

Androidのブラウザ（Chrome）で、UAにアクセスします。

※URL https://[UAのFQDN]/[UAの名前]/ua

ログイン画面が表示されるので、ユーザIDとパスワードを入力しログインします。

ログインすると、ユーザ専用ページが表示されます。

[ダウンロード]をタップし、証明書ファイルのダウンロードをおこないます。

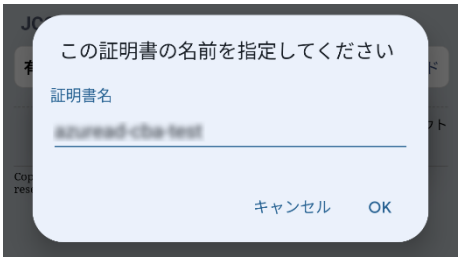
※「証明書 PIN」の値を「証明書を抽出」のパスワードとして入力します。

※インポートロックを有効にしている場合は、この時点からカウントが開始されます

[OK]をタップして続行してください。

「証明書の種類の用途」のダイアログが出るので、用途を選択します。

[OK]をタップして続行してください。

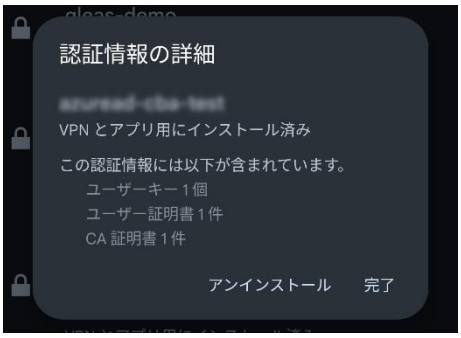


[OK]をタップします。

Chromeに戻り、[ログアウト]をタップしてUAからログアウトします。

以上で、Androidでの証明書ファイルのインストールは終了です。

[設定]>[セキュリティ]>[詳細設定]>[暗号化と認証情報]>[ユーザー認証情報]>[証明書の名前]とタップすると、インストールされた証明書情報を見ることができます。必要に応じて確認してください。



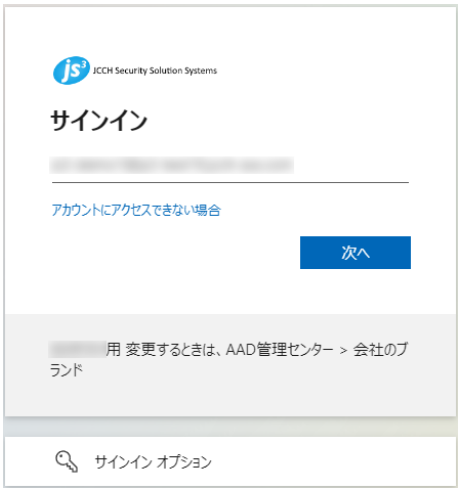
なお、インポートロックを有効にしている場合、[ダウンロード]をタップした時点より管理者の指定した時間を経過した後にUAに再ログインすると、以下の通り「ダウンロード済み」という表記に変わり、以後のダウンロードは一切不可となります。

8. 証明書ベース認証によるアプリケーション利用

8.1. Windows デバイスでアクセス

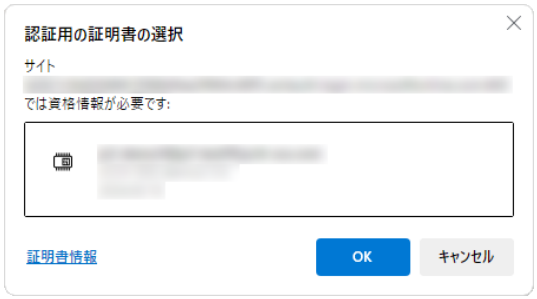
PCのブラウザ（Edge）から2.1項の「アプリケーションのサインオンURL」にアクセスすると、Entra ID のサインイン画面に遷移します。

ユーザー名を入力して次へをクリックします。



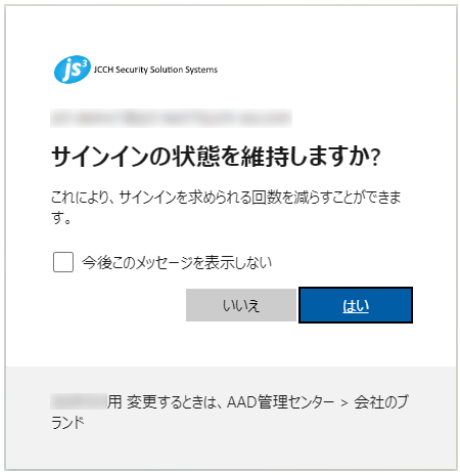
[証明書またはスマートカードを使用する]をクリックします。

クライアント証明書を選択して[OK]ボタンをクリックします。

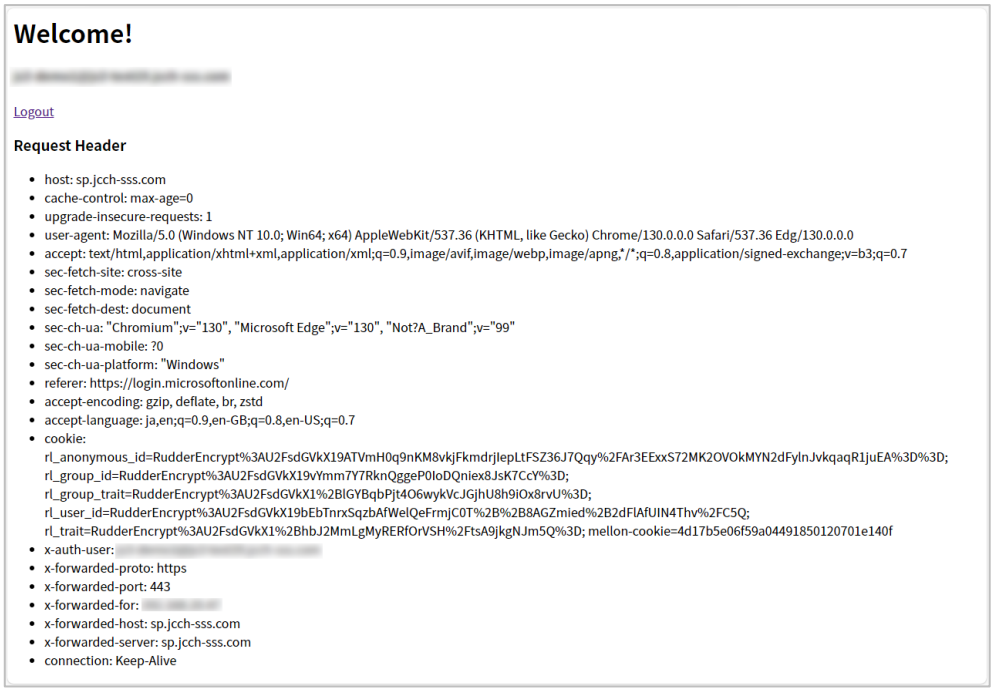


※パスワードを入力しても証明書の提示を強制されます。

[はい]または[いいえ]をクリックすると、サインインしてWebアプリケーションにアクセスできます。

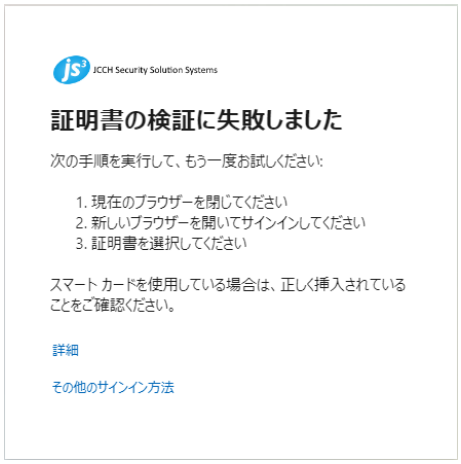


※以下は、アプリケーションにアクセスした例



証明書を持っていない場合や、失効済み証明書を提示した場合はアクセスに失敗します。

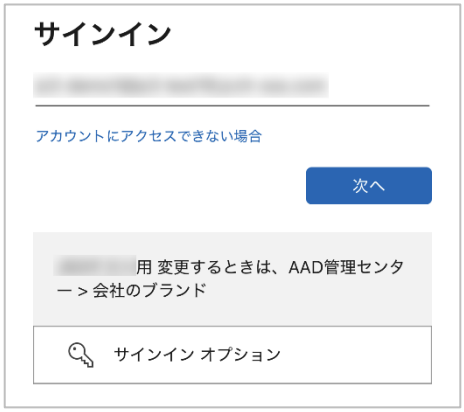
※以下は失効されたクライアント証明書でアクセスした例



8.2. iPhone デバイスでアクセス

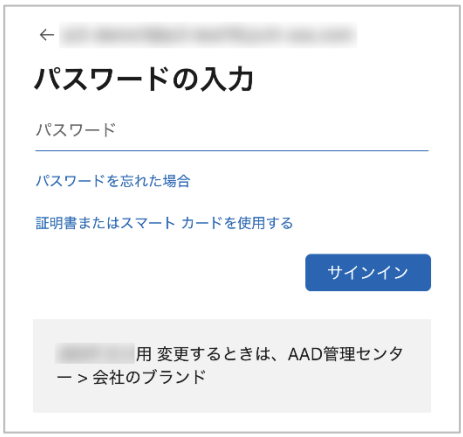
iPhoneのブラウザ（Safari）から2.1項の「アプリケーションのサインオンURL」にアクセスすると、Entra ID のサインイン画面に遷移します。

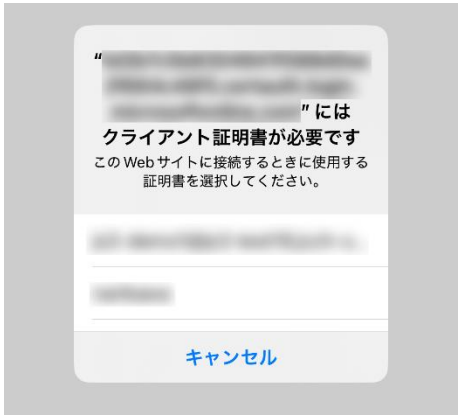
ユーザー名を入力して次へをタップします。



[証明書またはスマートカードを使用する]をタップします。

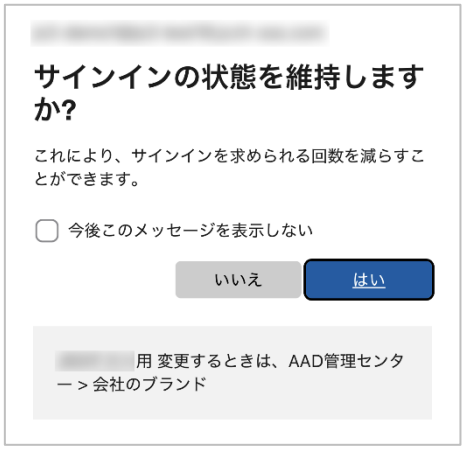
クライアント証明書を選択して[選択]をタップします。



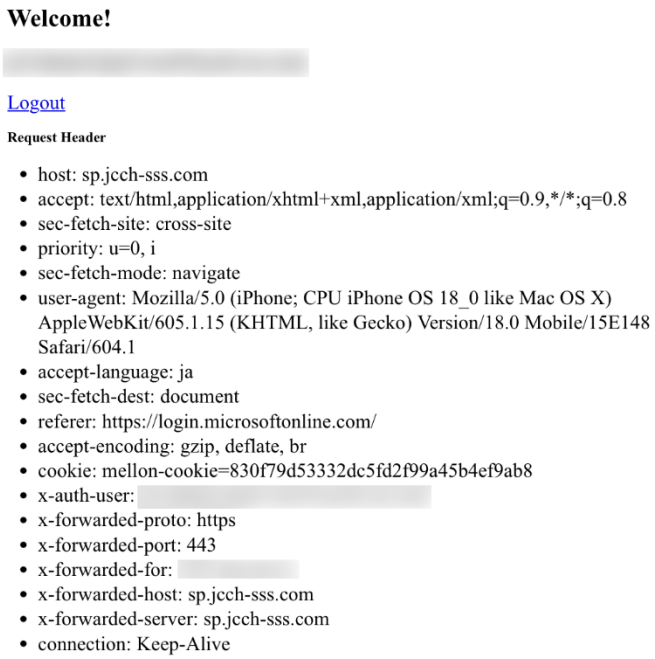


※デバイス内に証明書が1つのみ場合、証明書の選択画面はスキップ

[はい]または[いいえ]をタップすると、サインインしてWebアプリケーションにアクセスできます。

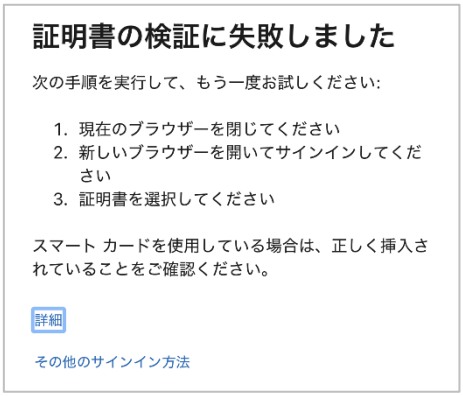


※以下は、アプリケーションにアクセスした例



証明書を持っていない場合や、失効済み証明書を提示した場合はアクセスに失敗します。

※以下は失効されたクライアント証明書でアクセスした例

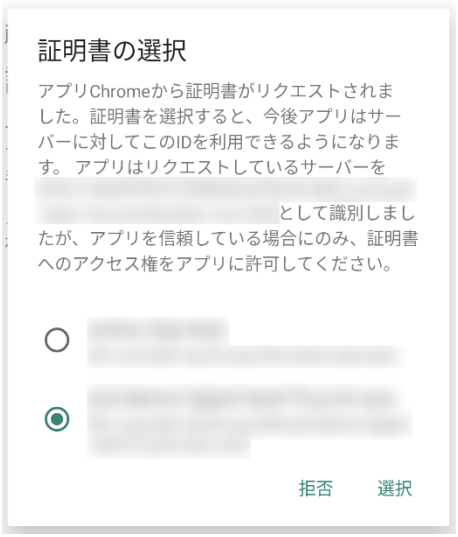


8.3. Android デバイスでアクセス

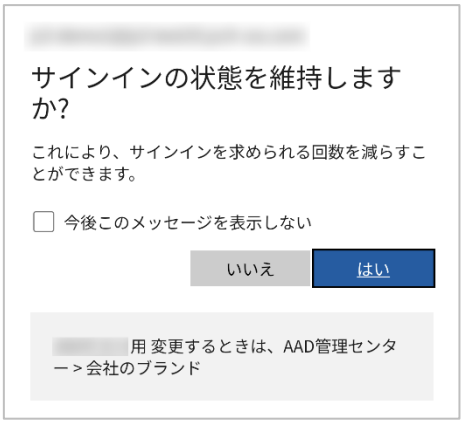
Adroidのブラウザ（Chrome）から2.1項の「アプリケーションのサインオンURL」にアクセスすると、Entra ID のサインイン画面に遷移します。

[ユーザー名]を入力して次へをタップします。

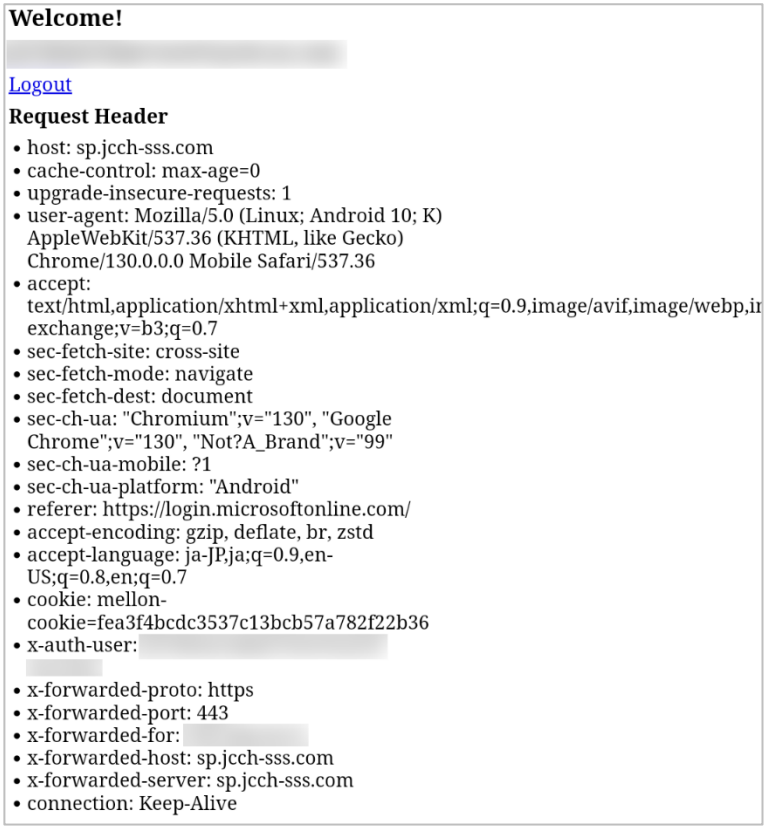
クライアント証明書を選択して[選択]をタップします。



[はい]または[いいえ]をタップすると、サインインしてWebアプリケーションにアクセスできます。

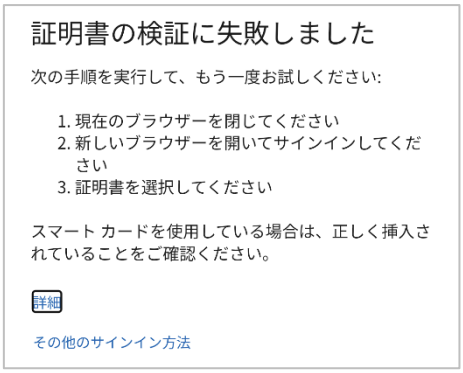


※以下は、アプリケーションにアクセスした例



証明書を持っていない場合や、失効済み証明書を提示した場合はアクセスに失敗します。

※以下は失効されたクライアント証明書でアクセスした例



9. その他

9.1. ユーザー情報をアプリケーションに伝える

本書の構成では、Entra ID ユーザー情報がアプリケーションに連携されます。

- Entra ID がSAML認証応答を発行

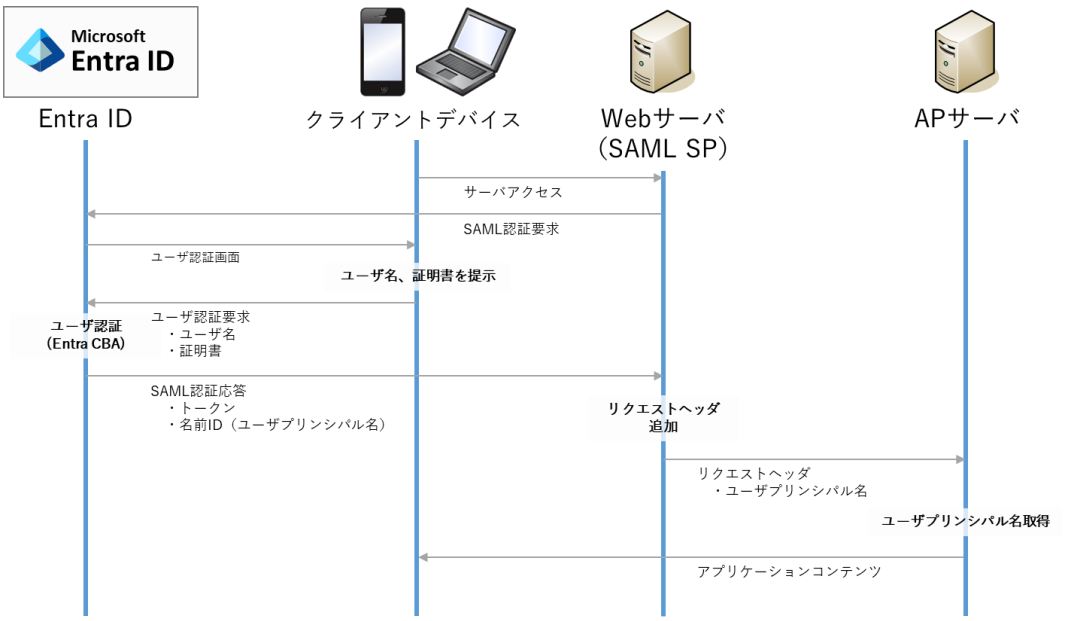
  認証が成功すると、Entra ID はSAML認証応答を発行します
  SAML認証応答のメッセージには認証済みユーザー情報が含まれます
  メッセージには名前IDとして Entra ID のユーザー プリンシパル名が記載されます
  ※その他 [追加の要求] として他のユーザー情報も送信可能です（3.5.6項の「属性とクレーム」）

- WebサーバがSAML認証応答を受け取る

  WebサーバはSAML認証応答から取り出したユーザー情報をリクエストヘッダに追加します
  リクエストヘッダを追加したHTTPリクエストをアプリケーションにリバースプロキシします
  ※設定例は 2.5 項を参照

- アプリケーションがユーザー情報を受け取る

  リクエストヘッダからユーザー情報を取り出すことができます
  ※実装例は 5.2 項を参照



9.2. サインインログについて

証明書ベース認証の状況は、Entra ID のサインイン ログから確認することができます。

Microsoft Entra 管理センター にログインします。

メニュー [アプリケーション] > [エンタープライズ アプリケーション]を選択します。

[すべてのアプリケーション] から対象のアプリケーションを選択します。

[サインイン ログ] をクリックするとログが一覧表示されます。

一覧からは以下を確認することができます。

ユーザー	サインインしたユーザー
アプリケーション	アクセスしたアプリケーション
状態	成功：認証が成功したログ 失敗：認証が失敗したログ 中断：証明書を要求しているログ
IPアドレス	アクセス元IPアドレス

証明書を要求しているログは [状態] が中断となっているのでログを選択すると、

[アクティビティの詳細:サインイン] から以下を確認できます。

基本情報	許可された時刻	アクセス日時
	ユーザー	アクセスしたユーザー
	アプリケーション	アクセスしたアプリケーション
場所	IPアドレス	アクセス元IPアドレス
デバイス情報	ブラウザ	アクセスブラウザの種類
	オペレーティングシステム	アクセスOSの種類
認証の詳細	認証方法	証明書認証のときは X.509 Certificate
	成功	true なら認証成功
追加の詳細	ユーザー証明書の***	認証時に提示された証明書情報

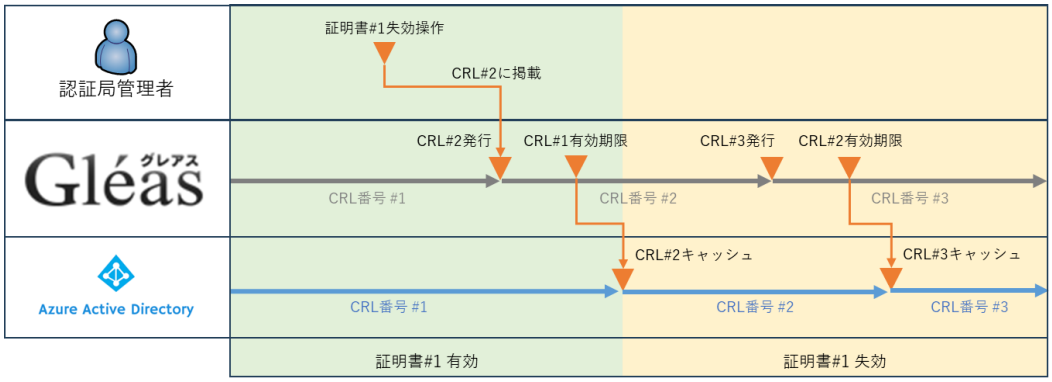
9.3. 証明書の失効確認について

証明書ベース認証は、証明書の失効を確認するために証明書失効リスト(CRL)を使用します。（OCSPはサポートされていません）

3.1項で登録した［証明書失効リストのURL］からCRLをダウンロードしてキャッシュします。CRLの有効期間が切れると最新のCRLを再ダウンロードしてキャッシュを更新します。

証明書ベース認証は、このキャッシュされたCRLを用いて失効確認を行うため、認証局での証明書失効操作は即時反映されません。

※失効操作が失効確認に反映されるまでのイメージ



9.4. 失効の即時反映について

証明書ベース認証では、手動操作による失効リストの即時反映が行えません。

利用者がデバイスを紛失したなどの理由で即時の失効が運用上必要な場合、Entra ID 上でユーザーの認証トークンを無効化することでアクセスを無効にする方法が考えられます。

認証を無効化してサインインを停止するコマンド例

- PowerShell を起動

  ※以降の操作の前に Microsoft Graph SDK をインストールが必要です。

```
Install-Module Microsoft.Graph
```

- 適切な資格情報で MgGraph サービスに接続

```
Connect-MgGraph -Scopes User.ReadWrite.All
```

- 認証トークンの無効化

```
Revoke-MgUserSignInSession -UserId [ユーザー プリンシパル名]
```

  ※この操作で現在の認証が無効化されます

- サインインを停止

```
Update-MgUser -UserId [ユーザー プリンシパル名] -AccountEnabled:$False
```

  ※この操作で指定ユーザーのサインインが禁止されます

9.5. 失効リストのサイズ制限について

証明書ベース認証で扱える失効リスト（CRL）は 20MB のサイズ制限があります。

失効リストに記載される失効情報は、該当証明書の有効期限まで記載され続けるため、

有効期間が長い証明書を運用する場合には失効リストの肥大化にもご留意ください。

10.問い合わせ

ご不明な点がございましたら、以下にお問い合わせください。

■Gléasや本検証内容、テスト用証明書の提供に関するお問い合わせ

株式会社JCCH・セキュリティ・ソリューション・システムズ

Tel: 050-3821-2195

Mail: sales@jcch-sss.com