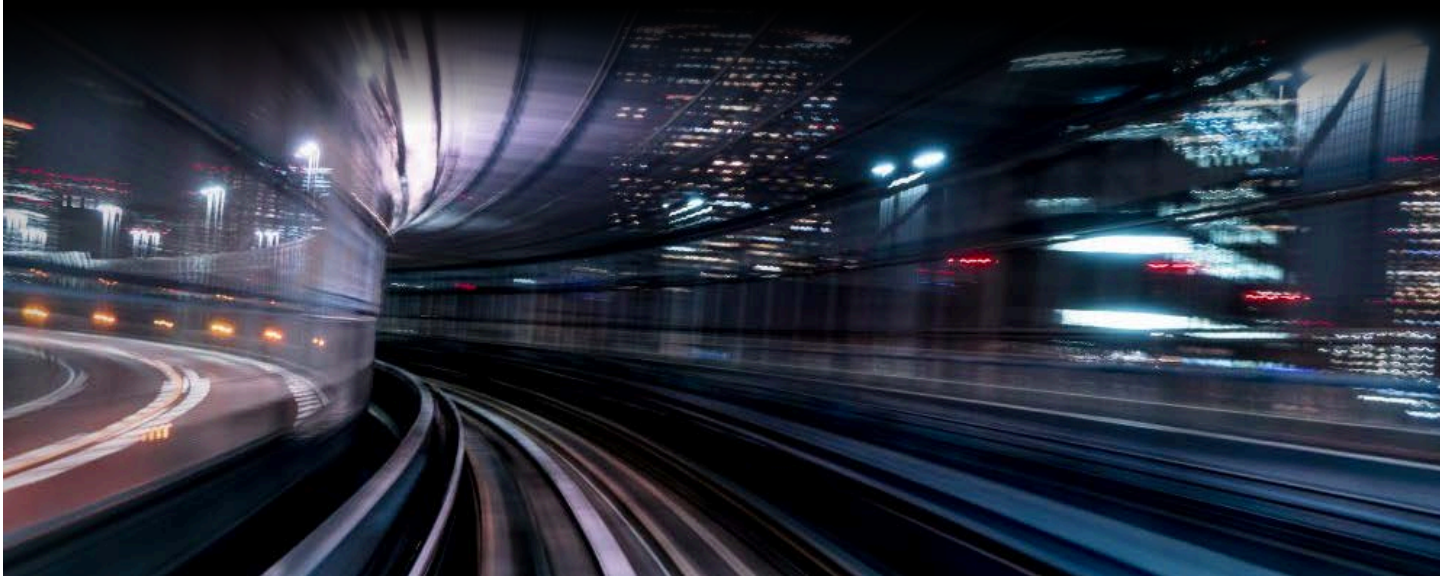
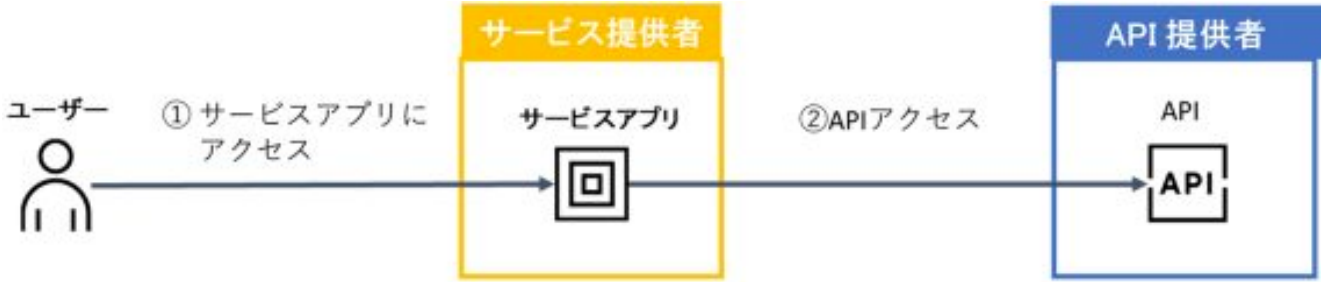
Azure API ManagementとIceWall Federation OIDC/OAuth OP ASとの連携

IceWall技術レポート



1. はじめに

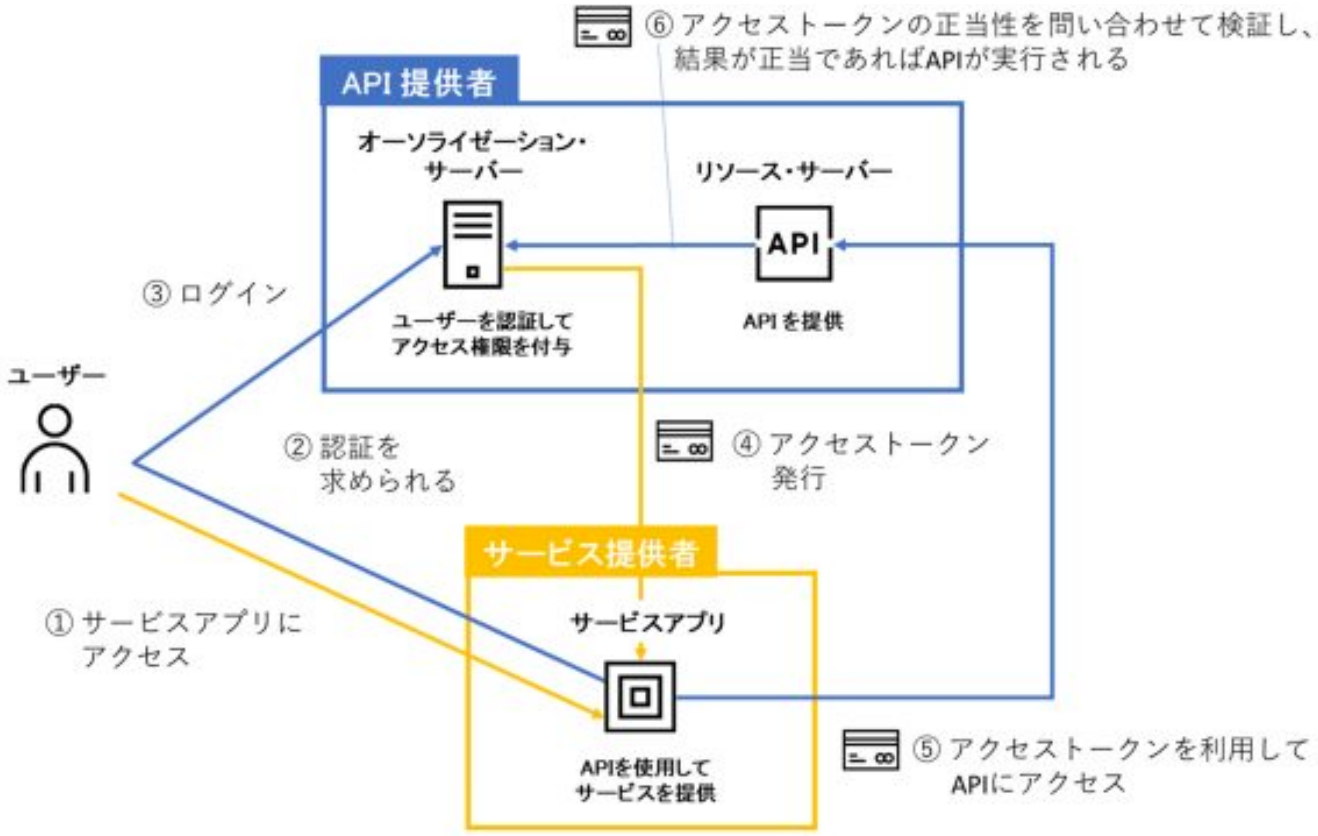
近年、企業が新規事業や更なるサービス展開のため、独自のAPIをWeb上に公開してユーザーに様々な情報を提供することが増えています。以下のように企業（API提供者）がAPIにより自社が持つデータを社外に公開することで、他の企業（サービス提供者）がそのデータを利用してユーザーに様々なサービスを提供することで可能となり、新しい形のサイトやビジネスが生まれるきっかけとなっています。



ただしAPI実行時はユーザーに代わりサービス提供者がユーザーの個人情報へアクセスするため、ユーザーの同意の取得やAPIにおけるアクセス認可をどう設定するかなど、APIサービス導入時にはセキュリティに関する課題があります。

そこでMicrosoft Azure が提供するAzure API ManagementとIceWall Federation OIDC/OAuth OP ASとを連携すると、セキュリティを考慮したAPIサービスの提供が可能となります。本技術レポートでは、その設定と動作検証についてご紹介します。

2.APIのアクセス認可



APIに直接アクセスするのはユーザーではなくサービス提供者ですが、ユーザーの認証はユーザーとAPI提供者の間で行います。ここで、ユーザーの認証はユーザーとAPI提供者の間で行った上でAPIへのアクセスをサービス提供者に許可するため、標準規格であるOAuthもしくはOpenID Connectを利用します。

APIのアクセス認可の処理の流れを以下に示します。ユーザーはサービスアプリにアクセスすると認証を求められ、「オーソライゼーション・サーバー」にてログインします。認証が完了すると、「オーソライゼーション・サーバー」は 「アクセストークン」を発行します。サービス提供者は「アクセストークン」を入手することで、APIへのアクセス権限を得ることができます。そしてサービスアプリが「アクセストークン」を利用してAPIへアクセスすると、「リソース・サーバー」が「オーソライゼーション・サーバー」に問い合わせて「アクセストークン」の正当性を検証し、結果が正当であればAPIが実行されます。

3. Azure API Managementとは

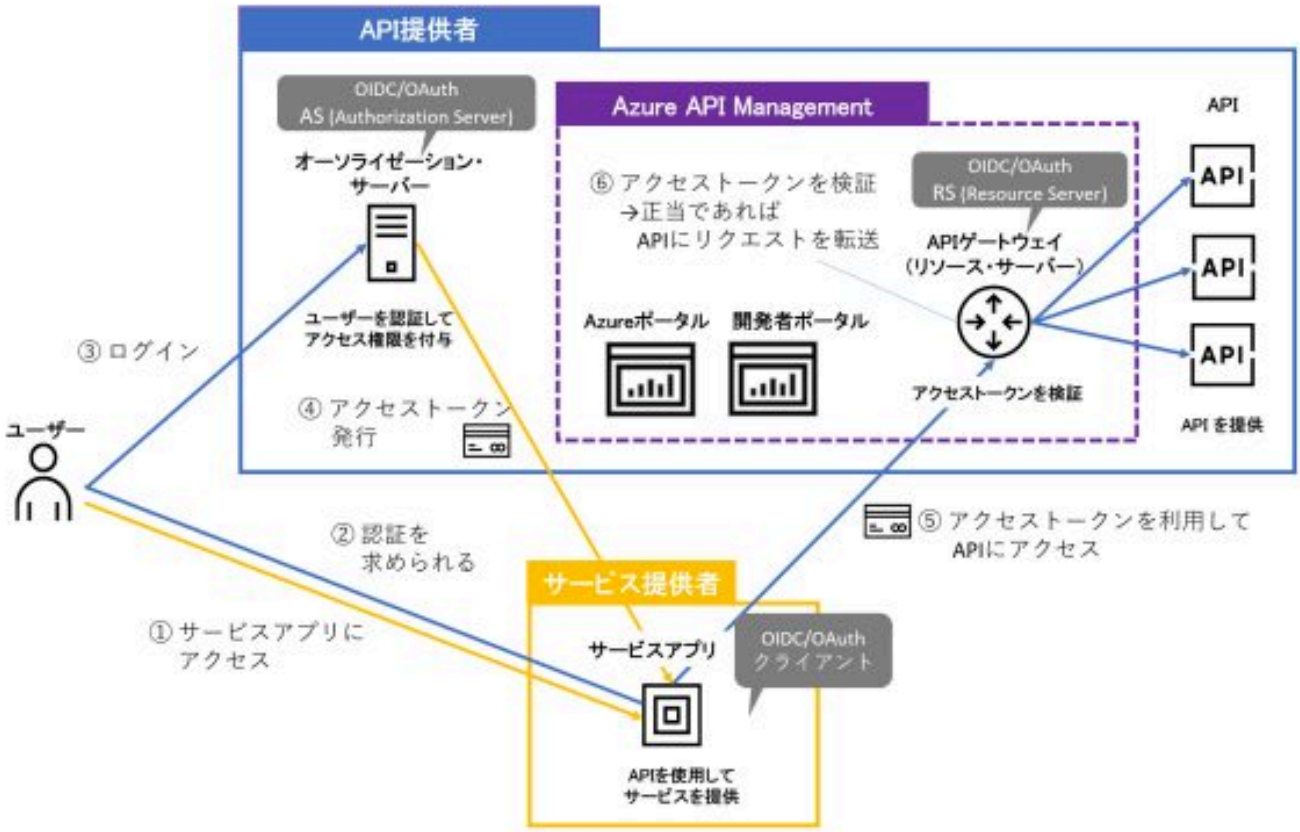
Azure API ManagementとはMicrosoft Azureが提供するAPIゲートウェイサービスであり、既存のAPIサービスをバックエンドとするAPIゲートウェイを迅速に作成することが可能です。API保護のためのセキュリティ設定や、APIの一元管理、API開発時のテスト環境など、API公開を全般的にサポートするサービスとなっています。

Azure API Managementは、エンドポイントとなる「APIゲートウェイ」、管理インターフェースとなる「Azure Portal」、API開発者のための「開発者ポータル」の3つの要素から構成されます。それぞれの役割を以下で説明します。

要素	概要	役割
APIゲートウェイ	エンドポイント	▪ API 呼び出しを受け入れ、バックエンドへルーティング ▪ API キー、JWT トークン、証明書、その他資格情報の検証 ▪ 使用量クォータとレート制限の適用 ▪ API 変換 ▪ バックエンドの応答キャッシュ ▪ 分析のための呼び出しメタデータの記録
Azure Portal	API プログラムをセットアップする管理インターフェイス	▪ API スキーマの定義またはインポート ▪ API を製品にパッケージ ▪ API のクォータや変換などのポリシー設定 ▪ 使用状況の分析 ▪ ユーザー管理
開発者ポータル	開発者用のメインWeb	▪ API のドキュメント閲覧 ▪ 対話型コンソールを使用したAPIの テスト ▪ API キー取得のためのアカウント作成・サブスクライブ ▪ API使用に関する分析

※ 詳細は公式ドキュメントをご参照ください。

Azure API Managementのドキュメント - API Management について
https://docs.microsoft.com/ja-jp/azure/api-management/api-management-key-concepts



Azure API Managementを使用した場合のAPIのアクセス認可の処理の流れを以下に示します。

Azure API Management を使用すると、APIのアクセス認可におけるアクセストークンの検証をAPIゲートウェイ上で行うことが可能なため、各APIでアクセストークン検証のための実装は必要ありません。

4. Azure API ManagementとIceWall Federation OIDC/OAuth OP ASの連携

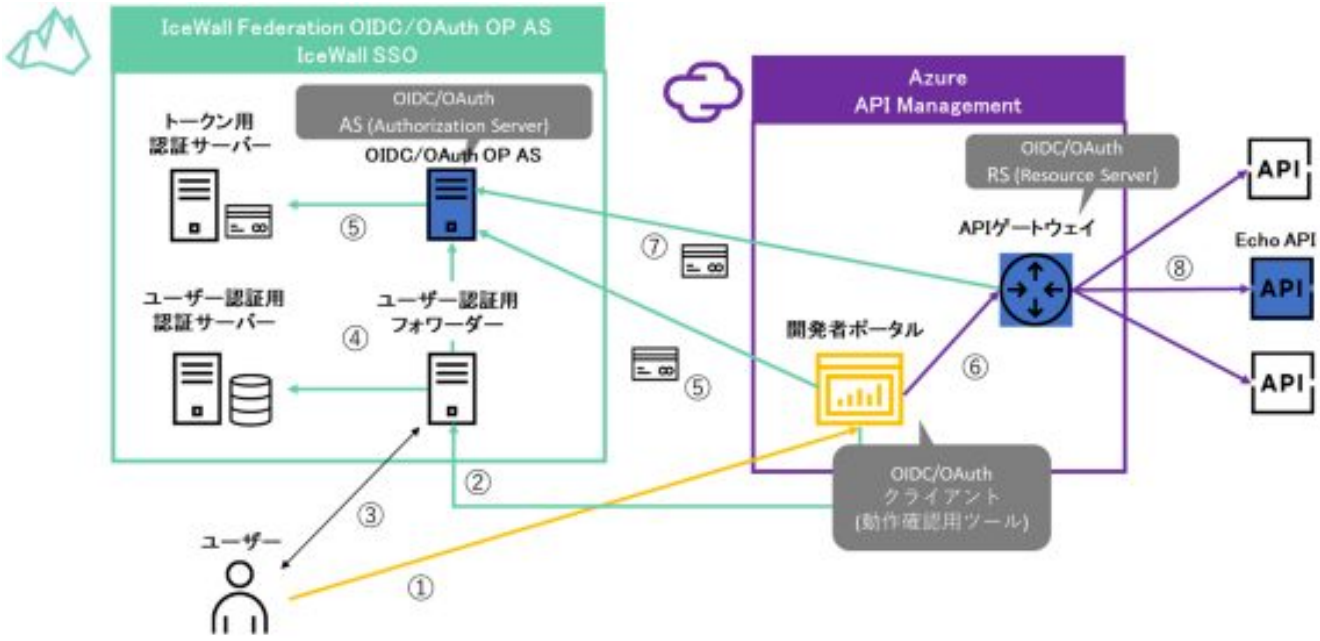
4.1 システム構成

Azure API ManagementとIceWall Federation OIDC/OAuth OP ASとが連携したシステムの概要は以下です。

API提供者のうち、リソース・サーバーにはAzure API ManagementのサンプルAPI "Echo API"を使用し、オーソライゼーション・サーバーはIceWall Federation OIDC/OAuth OP AS（以降IceWall OP サーバー）が担います。IceWall OP サーバーの前段にはIceWall SSO（ユーザー認証用フォワーダー、ユーザー認証用認証サーバー）を用意し、ユーザーの認証はIceWall SSOが行います。

サービス提供者のサービスアプリには、Azure API Management開発者ポータルにおいてAPI開発者が動作確認に使用可能なOIDC/OAuthクライアントツールを使用します。開発者ポータルではこの動作確認用OIDC/OAuthクライアントツール（以降OIDC/OAuthクライアント）を使用することで、クライアントアプリケーションを用意せずに（※）APIのテストを行うことができます。

※本番運用では別途クライアントアプリケーションが必要です。



①ユーザーがOIDC/OAuthクライアントにアクセスします。

②OIDC/OAuthクライアントがIceWall OP サーバーへアクセスします。

③ユーザー認証（ログイン）が未実施の場合、IceWall OP サーバー前段のユーザー認証用フォワーダーが認証を要求します。

④ユーザーがログインします。

⑤ログインが完了すると、OIDC/OAuthクライアントはOAuthプロトコルを使用して、IceWall OP サーバーからアクセストークンを取得します。

⑥OIDC/OAuthクライアントがアクセストークンを付加して、APIへのアクセス要求をAPIゲートウェイに送信します。

⑦APIゲートウェイがIceWall OP サーバーのToken Introspection Endpointへアクセストークンの検証を依頼します。

⑧検証結果が正当であれば、APIゲートウェイはAPIへリクエストを転送します。

4.2 Azure API Managementの設定

Microsoft Azure PortalにおいてAPI Managementサービスの設定を行います。

4.2.1 API Managementサービスの作成

まずAPI Managementサービスを作成します。「リソースの作成」から「API Management」を検索し、API Managementサービスを新規作成します。

必要事項を記入し「作成」をクリックすると、リソースグループへのデプロイメントが開始されます。デプロイメントが完了しオンライン（使用可能）状態となると、設定を開始することができます。

4.2.2 OPの登録

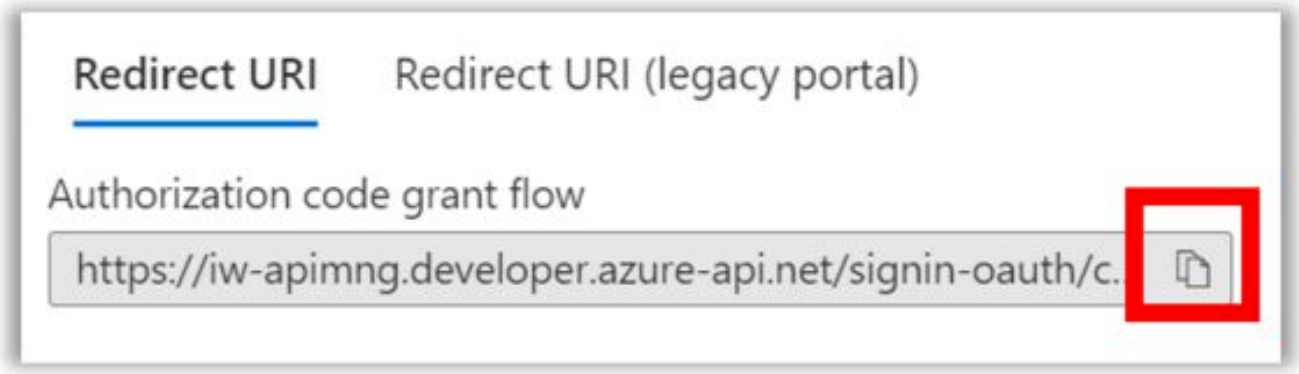
OIDC/OAuthクライアントに割り当てるOP (OpenID Provider)を登録します。

登録可能なプロトコルにはOpenID ConnectとOAuth 2.0が用意されており、クライアントは取得したIDトークンもしくはアクセストークンをAPIに送信します。ここではAPIゲートウェイがIceWall OP サーバーのToken Introspection Endpointを使用したアクセストークンの検証を行うため、OAuth 2.0を選択してIceWall OP サーバーを登録します。

作成したAPI Managementサービス内メニューの「開発者ポータル – OAuth2.0」にて、「追加」をクリックします。

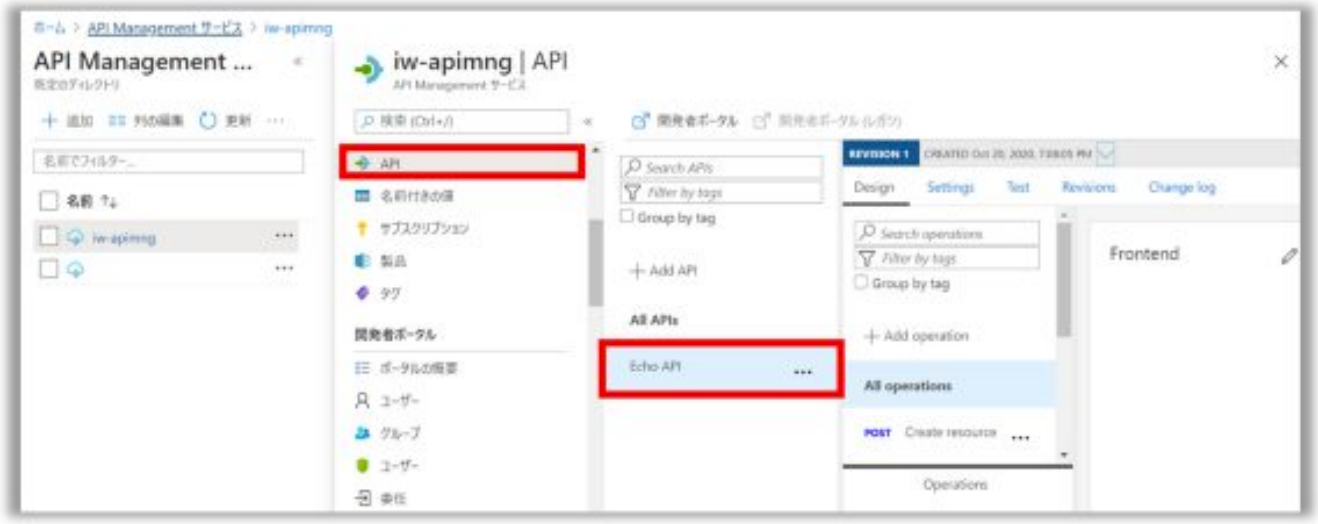
以下の必要事項を記入の上「保存」をクリックし、OPの設定を登録します。

項目名	設定値
表示名	任意の名前
ID	任意の名前
説明	任意の内容
承認許可の種類	承認コード
承認エンドポイントのURL	https://< FQDN >/fw/dfw/OP/op/auth ※ fw :前段フォワーダーのスクリプトエイリアス OP : IceWall OP サーバーのエイリアス名
承認要求方法	GET
トークンエンドポイントのURL	https://< FQDN >/fw/dfw/OP/op/token
クライアント認証方法	本文内
アクセストークンの送信方法	Authorizationヘッダー
規定のスコープ	sample_scope (ダミーの値)
クライアントID/クライアントシークレット	任意の値



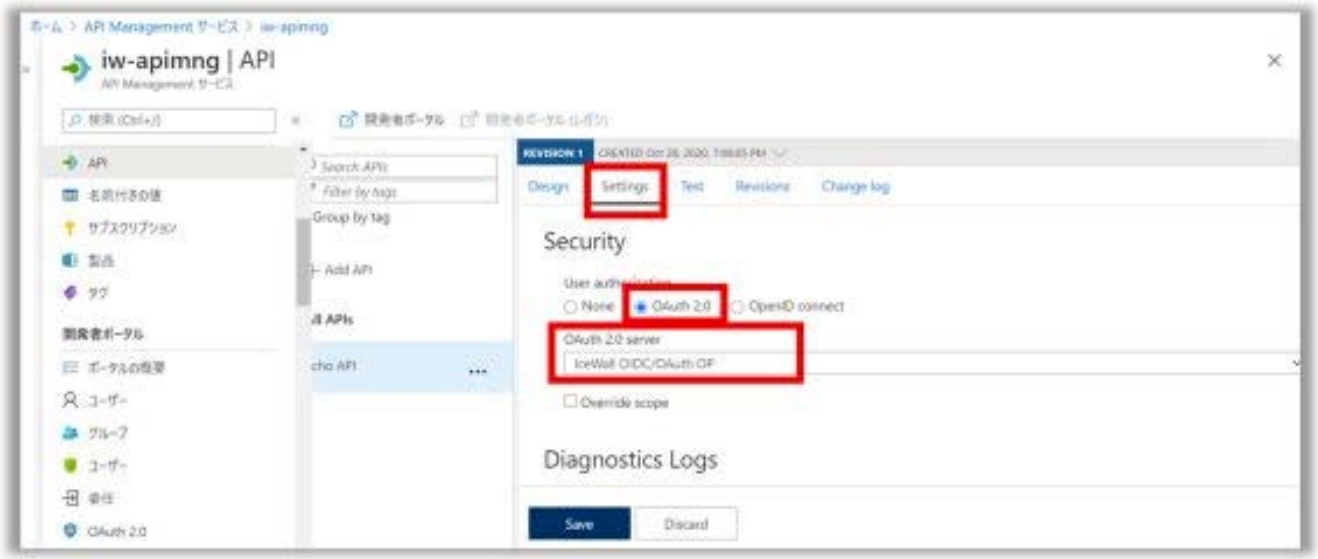
「既定のスコープ」「クライアントID/クライアントシークレット」「Redirect URI」はIceWall OP サーバーにも同じ値を設定します（4.3.1節）。「Redirect URI」はコピーしておくことができます。

4.2.3 APIへのOPの割り当て



テストに使用する”Echo API”のユーザー認証設定に、先ほど登録したOPを割り当てます。

API Managementサービス内メニューの「APIs – API」にて ”Echo API”をクリックします。



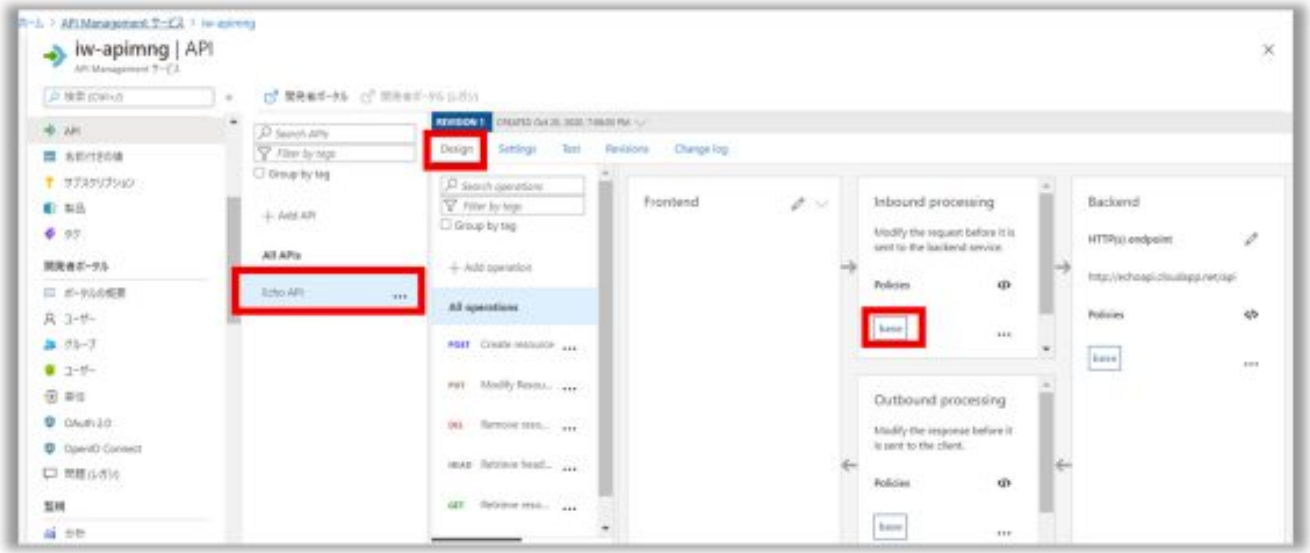
「Settings」タブ内の 「Security」 項目にて先ほど登録したOPを選択し、「Save」をクリックして保存します。

4.2.4トークン検証ポリシーの設定

Azure API ManagementにおけるAPIの動作を制御可能な「ポリシー」機能を使用して、APIゲートウェイがAPIへのリクエストを受信した際にアクセストークンを検証するよう設定します。アクセストークンの検証にはIceWall OP サーバーに用意されているToken Introspection Endpointを使用します。設定内容は公式ドキュメントのポリシー設定例（※）を参考にします。

※Azure API Managementのドキュメント - Azure API Management サービスからの外部サービスの使用

https://docs.microsoft.com/ja-jp/azure/api-management/api-management-sample-send-request



API Managementサービス内メニューの「APIs – API」にて、”Echo API”をクリックします。
「Design」タブ内の「Inbound processing」をクリックして、インバウンドのポリシーを編集します。

```
<inbound>
        <set-variable name="token"
value="@(context.Request.Headers.GetValueOrDefault("Authorization","scheme param").Split(' ').Last())" />
        <send-request mode="new" response-variable-name="tokenstate" timeout="20" ignore-error="true">
            <set-url>https://<FQDN>/fw/dfw/OP/op/internal/admin_token_introspection</set-url>
            <set-method>POST</set-method>
            <set-header name="Authorization" exists-action="override">
                <value>Basic <"ユーザー名:パスワード"をBase64化した文字列></value>
            </set-header>
            <set-header name="Content-Type" exists-action="override">
                <value>application/x-www-form-urlencoded</value>
            </set-header>
            <set-body>@($"token={(string)context.Variables["token"]}")</set-body>
        </send-request>
        <choose>
            <when
condition="@((bool)((IResponse)context.Variables["tokenstate"]).Body.As<JObject>()["active"] == false)">
                <return-response response-variable-name="existing response variable">
                    <set-status code="401" reason="Unauthorized" />
                    <set-header name="WWW-Authenticate" exists-action="override">
                        <value>Bearer error="invalid_token"</value>
                    </set-header>
                </return-response>
            </when>
        </choose>
        <base />
    </inbound>
```

IceWall OPサーバーではToken Introspection Endpointへアクセス可能なリクエストをAPI ゲートウェイからのリクエストに制限するため、Token Introspection Endpointへのリクエストにはベーシック認証を求めるよう設定します（4.3.2節）。そこでAuthorizationヘッダーのvalueには、そのベーシック認証で許可する「ユーザー名：パスワード」をBase64化した文字列を設定します。

4.3 IceWallの設定

IceWallはOPサーバーのセット（OIDC/OAuth OP AS、トークン用認証サーバー）と、IceWall SSOのセット（ユーザー認証用フォワーダー、ユーザー認証用認証サーバー）から構成されます。

4.3.1 IceWall OPサーバーの設定

クライアント設定ファイルに以下を設定します。「client_id」「client_secret」項目はAzure API Managementで設定したクライアントID、クライアントシークレットを、「redirect_uris」項目にはAzure API Managementで指定された値を設定します。

また今回はOAuthで動作させるため、オブジェクト設定ファイルを新規作成してダミーのスコープを設定し、「scopes_supported」項目に設定します。

■クライアント設定ファイル

```
client_id=< API Managementで設定したクライアントID>
client_secret=< API Managementで設定したクライアントシークレット>
redirect_uris=< Azure API Managementで指定された値 >
response_types_supported=code
grant_types_supported=authorization_code
scopes_supported= sample_scope
```

■オブジェクト設定ファイル

```
scope=sample_scope
```

4.3.2 ユーザー認証用フォワーダーApacheの設定

IceWall OP サーバーのAuthorization Endpointへのリクエストにはnonceパラメーターの付与が必須（※）ですが、OIDC/OAuthクライアントツールではリクエストにnonceパラメーターが付与されません。

※将来のパッチでnonceパラメーターはオプションとなる予定です。

そこでユーザー認証用フォワーダーのApacheにて、Authorization Endpointへのアクセスにはnonceパラメーターを付与するよう設定します。

```
RewriteEngine on
RewriteCond %{HTTPS} on
RewriteCond %{REQUEST_URI} ^/fw/dfw/OP/op/auth$
RewriteCond %{QUERY_STRING} !(^|&)nonce=
RewriteRule ^.*$ %{REQUEST_URI}?nonce=123 [QSA,R=302,NE]
```

また、IceWall OP サーバーのToken Introspection Endpointへアクセス可能なリクエストをAPI ゲートウェイからのリクエストに制限するため、Token Introspection Endpointへのリクエストにはベーシック認証を求めるよう設定しておきます。

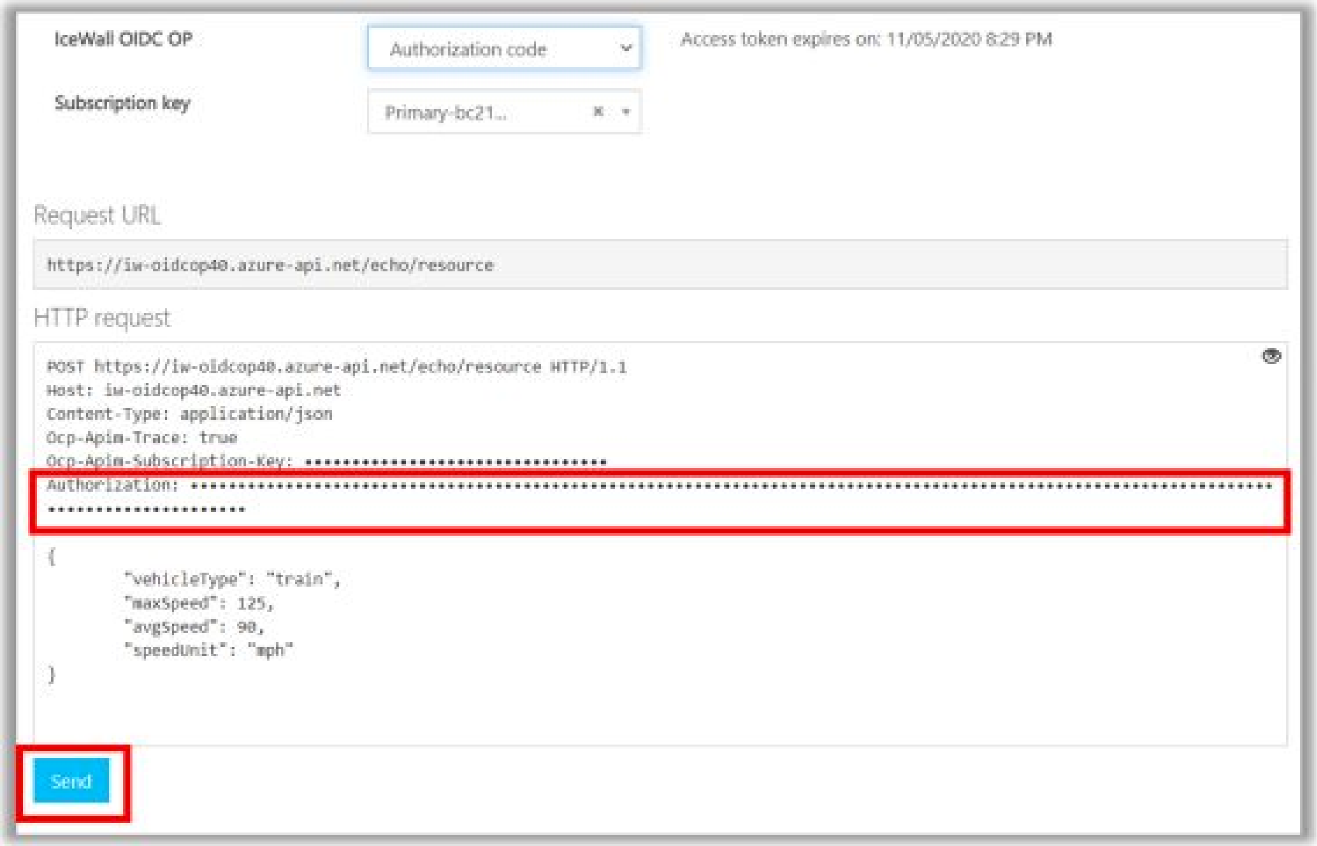
5.動作確認

Azure API ManagementとIceWall Federation OIDC/OAuth OP ASとが連携したシステムによる、APIのアクセス認可の動作確認を行います。

1.「開発者ポータル - ポータルの概要」から「開発者ポータル（レガシ)」をクリックします。

2.開発者ポータルが起動したら、「APIs」タブから”Echo API”を選択するとAPIドキュメントが表示されます。任意のoperationを選択し「Try IT」ボタンをクリックすると、API実行画面が表示されます。

3.「Authorization」 選択項目にて、IceWall OP サーバーの「Authorization Code」を選択します。



4.別ウィンドウでIceWall OP サーバーの認証画面が表示されます。

5.認証、認可同意が完了すると別ウィンドウが閉じて開発者ポータル画面が読み込まれ、APIへのリクエスト電文中のAuthorizationヘッダーにアクセストークンがセットされます。

6.「Send」をクリックしてAPIにリクエストを送信します。

7.リクエストを受け取ったAPIゲートウェイはポリシーに基づいてIceWall OP サーバーのToken Introspection Endpointへアクセスし、アクセストークンの検証を依頼します。

8.アクセストークンの検証に成功するとAPIが実行され、APIから200 OKが返ります。

6. まとめ

Azure API ManagementとIceWall Federation OIDC/OAuth OP ASを連携することで、セキュアなAPIサービスを提供することが可能です。今回はその手順をご紹介しました。
APIをクラウド環境で公開される際は、今回のシステムをぜひご検討ください。

2020.11.20

執筆者　：日本ヒューレット・パッカード株式会社

Pointnext事業統括　認証コンサルティング部

藤 ひとみ

技術レポート一覧へ →

お問い合わせ先一覧

企業情報

会社情報

アクセス

お問い合わせ

採用情報

HPEについて

インクルージョン & ダイバーシティ

サステナビリティと企業責任

経営幹部

お知らせ

ニュースルーム

イベント・セミナー

新着情報

重要なお知らせ

パートナー

パートナープログラム

認定資格制度

OEMソリューション

サポート

製品サポート

ソフトウェア & ドライバー

標準保証確認

オペレーショナルサポート

教育とトレーニング

製品リサイクル

機器部品の妥当性確認

コミュニティ

HPE Japan ブログ

リソース

お客様事例

ご購入方法

オンラインストア

HPE Customer Center

Eメール登録

ドキュメントライブラリ

Resource Library

ビデオギャラリー

金融サービス

日本 (ja)