

# Amazon API Gateway 連携ガイド

2022. **3**  
March

このガイドには、次の表に示す変更履歴が含まれています。

作成年月	変更内容	変更区分(*1)
2022年3月	初版	—

なお、単なる誤字・脱字などは、お断りなく訂正しました。

注 \*1

変更区分 C：案内を変更（変更又は削除）します。既存のユーザは、使い方を変更する必要があります。

変更区分 A：既存のユーザには影響ありません。新しい案内を適用する場合だけ、使い方を変更する必要があります。

変更区分 S：案内の変更はありません。説明の追加・変更があります。

# はじめに

このドキュメントは、uCosminexus Service Platform で作成・公開した REST API を、Amazon API Gateway（以下 API Gateway）を利用して管理・運用するために、最初に行う設定について、解説したものです。

Amazon Web Services（AWS）および API Gateway の詳細については、AWS の Web サイトを参照してください。

## ■商標類

- ・ Amazon Web Services、『Powered by Amazon Web Services』ロゴ、Amazon EC2、Amazon VPC、AWS は、米国および/またはその他の諸国における、Amazon.com, Inc. またはその関連会社の商標です。
- ・ その他記載の会社名、製品名は、それぞれの会社の商標もしくは登録商標です。

## ■AWS のサービス略称

AWS のサービス名称の略称は次の通りです。詳細は AWS のウェブサイト (<http://aws.amazon.com>) を参照してください。

サービス略称	AWS のサービス名称
Amazon EC2	Amazon Elastic Compute Cloud
Amazon VPC	Amazon Virtual Private Cloud

## ■このガイドをご使用いただく際の注意点

このガイドに記載されている仕様は、製品の改良により予告なく変更されることがあります。

## ■発行元

株式会社日立製作所 サービス&プラットフォームビジネスユニット  
サービスプラットフォーム事業本部

All Rights Reserved. Copyright (C) 2022, Hitachi, Ltd.

# 目 次

<b>1 API Gateway 連携について</b>	<b>1</b>
1.1 API Gateway とは.....	2
<b>2 API Gateway の設定・利用について</b>	<b>3</b>
2.1 API Gateway の作成.....	5
2.2 API Gateway の設定.....	6
2.2.1 アクセス用インターフェイス VPC エンドポイントの設定.....	6
2.2.2 呼び出し先 VPC リンクの作成.....	7
2.2.3 統合リクエストの設定.....	8
2.2.4 ステージの設定.....	10
2.2.5 オーソライザーの設定.....	11
2.3 API Gateway の利用.....	12
2.3.1 API Gateway の呼び出し.....	12
2.3.2 ダッシュボードの利用.....	13

# 1 API Gateway 連携について

この章では、API Gateway 連携について説明します。

本章の構成

## 1.1 API Gateway とは

## 1.1 API Gateway とは

---

API Gateway は、AWS で提供される、フルマネージド型 API 統合サービスです。uCosminexus Service Platform で作成した REST API を管理・公開したり、認証などの機能を追加したりできます。

# 2 API Gateway の設定・利用について

この章では、API Gateway の設定・利用について説明します。

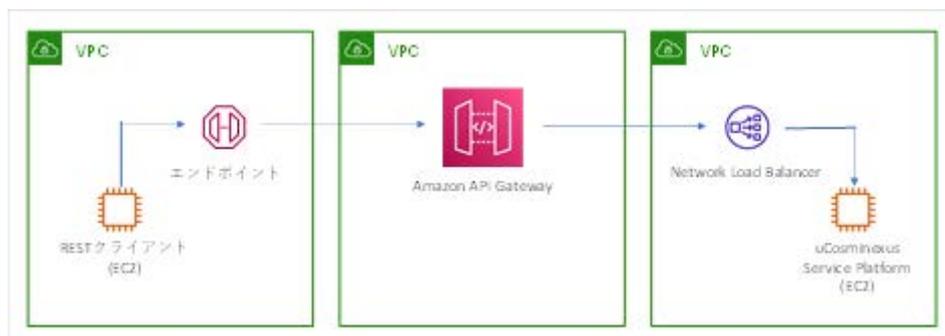
本章の構成

2.1 API Gateway の作成

2.2 API Gateway の設定

2.3 API Gateway の利用

このドキュメントの説明では、プライベートサービス(ローカル IP アドレスを利用したサービス)の作成について解説しています。このドキュメントで説明するシステムの構成図を次に示します。



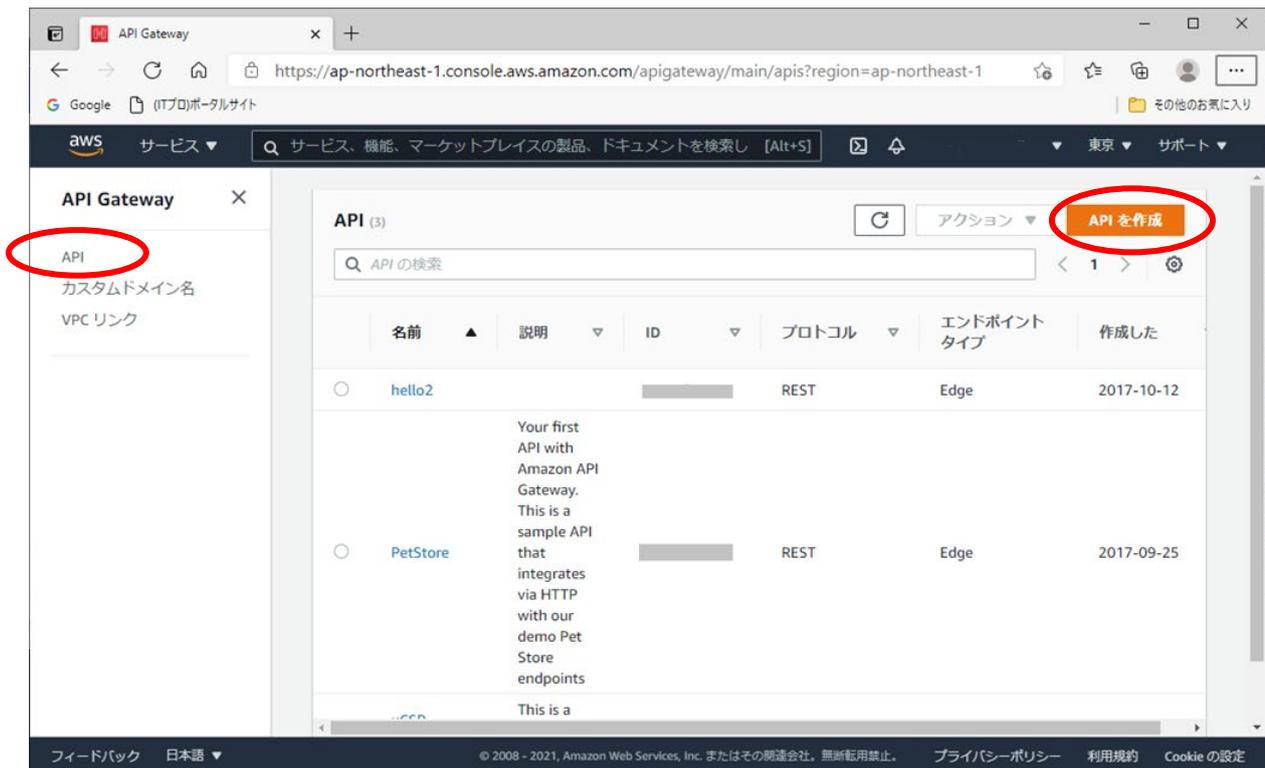
呼び出し先の REST API は、uCosminexus Service Platform 上で作成され、Amazon EC2 にデプロイしてあるものとします。また、API Gateway で公開する API の OpenAPI ドキュメントも作成してあるものとします。

## 2.1 API Gateway の作成

API Gateway を利用するために、API の作成をします。

AWS マネジメントコンソールにサインインします。その後、サービスの一覧から API Gateway を選択します。

API Gateway の画面で、左メニューから「API」を選択します。その後「API を作成」を選択します。



「API タイプを選択」画面では、「REST API プライベート」のインポートを選択します。

「新しい API の作成」画面では、「Swagger あるいは Open API 3 からインポート」が選択されていることを確認して、あらかじめ作成しておいた YAML (または JSON) 形式の API 定義ファイルを指定します。エンドポイントタイプが「プライベート」であることを確認し、「インポート」を選択します。

API がインポートされ、作成が完了します。

## 2.2 API Gateway の設定

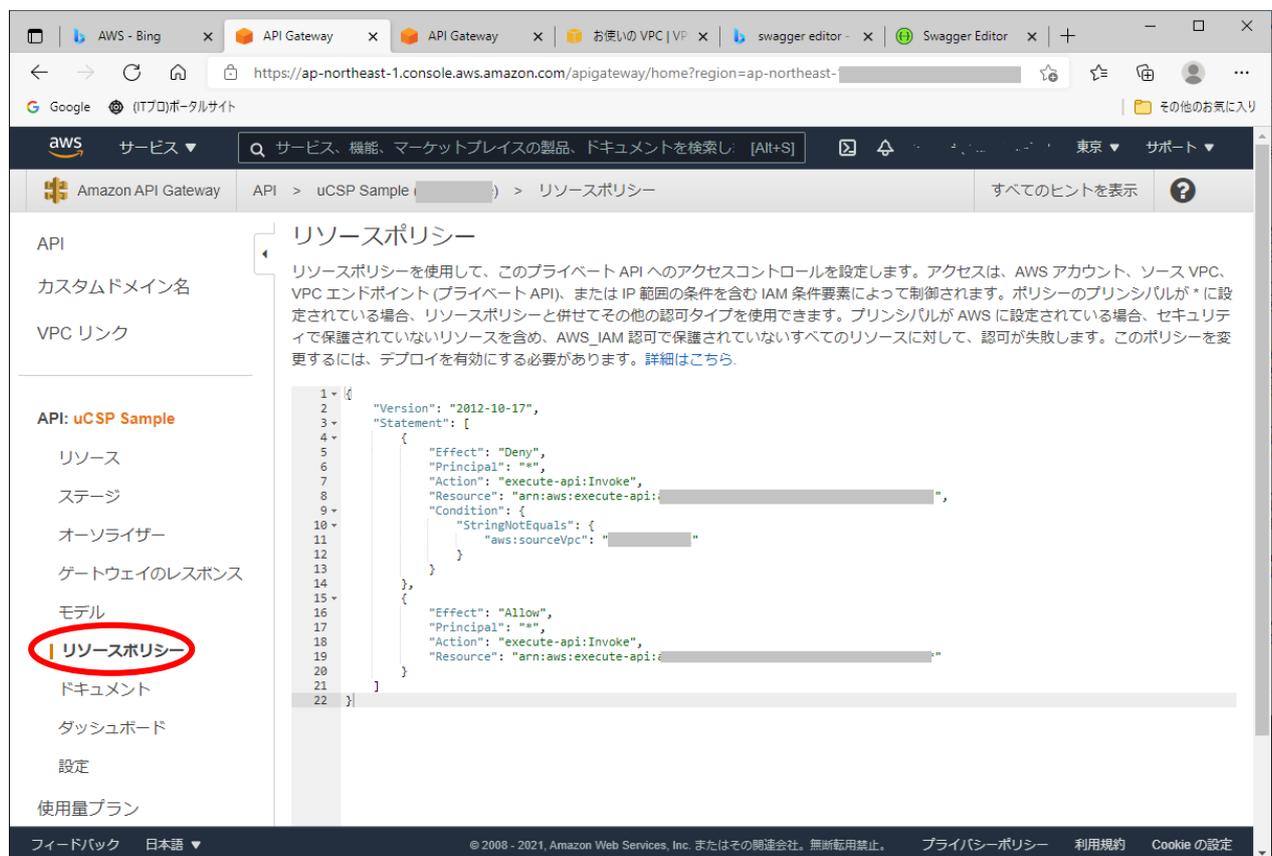
### 2.2.1 アクセス用インターフェイス VPC エンドポイントの設定

作成された API Gateway は、独自の VPC に所属しているため、クライアントからアクセスするには API Gateway 用のインターフェイス VPC エンドポイントを経由する必要があります。

インターフェイス VPC エンドポイントにアクセスするためには、API Gateway 用のインターフェイス VPC エンドポイントのための API Gateway リソースポリシーを作成し、API にアタッチする必要があります。

次の例では、(クライアントの存在する)呼び出し元になる VPC を指定し、API へのアクセスを許可しています。

API を選択し、左メニューから「リソースポリシー」を選択します。



The screenshot shows the AWS API Gateway console interface. The left-hand navigation menu is visible, with 'リソースポリシー' (Resource Policy) highlighted in red. The main content area displays the 'リソースポリシー' configuration for the 'API: uCSP Sample'. The policy is defined in JSON format, showing a 'Deny' statement for requests from a specific VPC and an 'Allow' statement for all other requests.

```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Effect": "Deny",
6        "Principal": "*",
7        "Action": "execute-api:Invoke",
8        "Resource": "arn:aws:execute-api:...",
9        "Condition": {
10       "StringNotEquals": {
11         "aws:sourceVpc": "..."
12       }
13     }
14   },
15   {
16     "Effect": "Allow",
17     "Principal": "*",
18     "Action": "execute-api:Invoke",
19     "Resource": "arn:aws:execute-api:..."
20   }
21 ]
22 }
```

リソースポリシーを作成し、保存します。

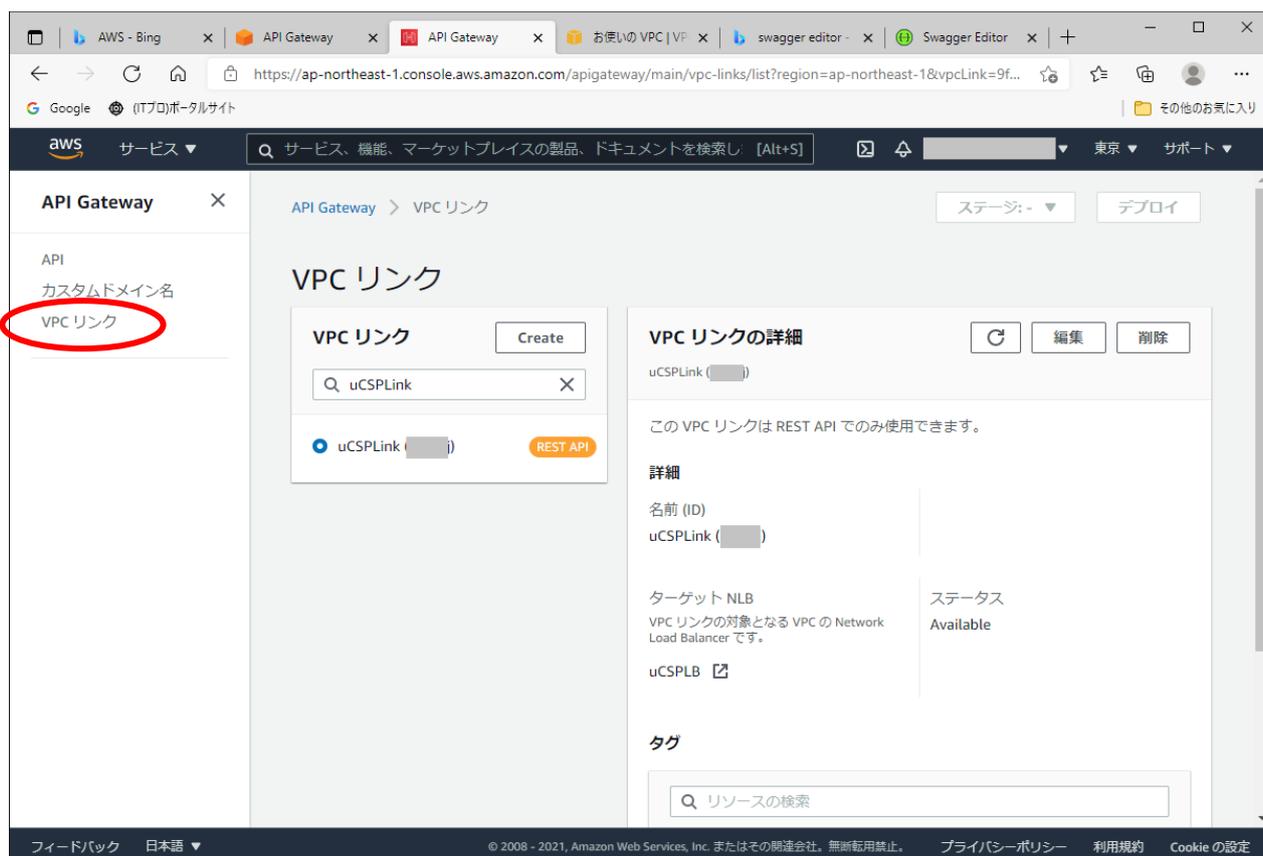
## 2.2.2 呼び出し先 VPC リンクの作成

プライベート API Gateway から呼び出し先のバックエンド API を呼び出すには、API Gateway の統合リクエストで VPC リンクを作成・指定する必要があります。

VPC リンクのターゲットには、Network Load Balancer を指定できます。

最初に、呼び出し先の uCosminexus Service Platform を含んだ Network Load Balancer を作成します。その後、VPC リンクを作成し、Network Load Balancer を指定します。

API Gateway の画面の左メニューから、「VPC リンク」を選択します。

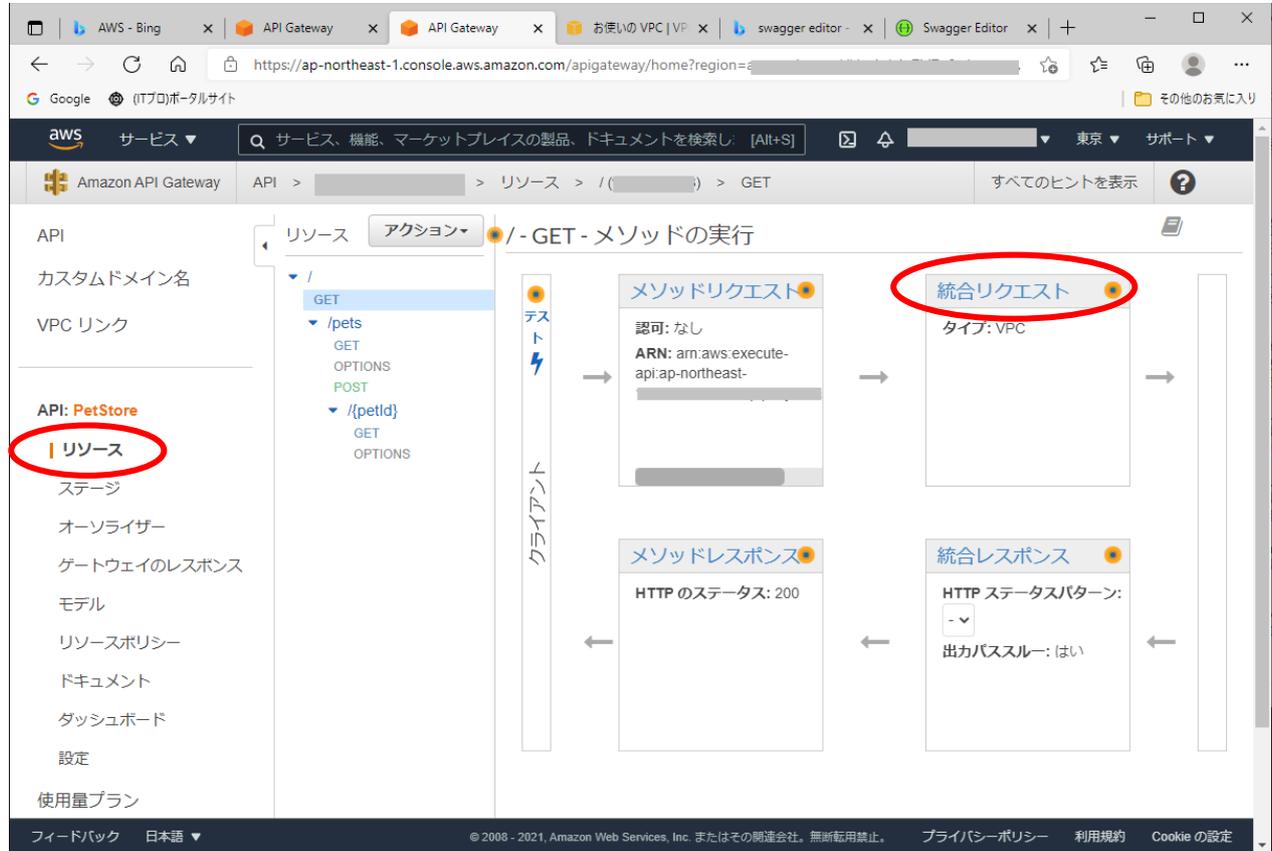


作成する VPC リンクの名称を決定し、ターゲット NLB として、作成した Network Load Balancer を指定します。上記の例では、「uCSPLB」という名称の Network Load Balancer に接続する VPC リンクを作成しています。

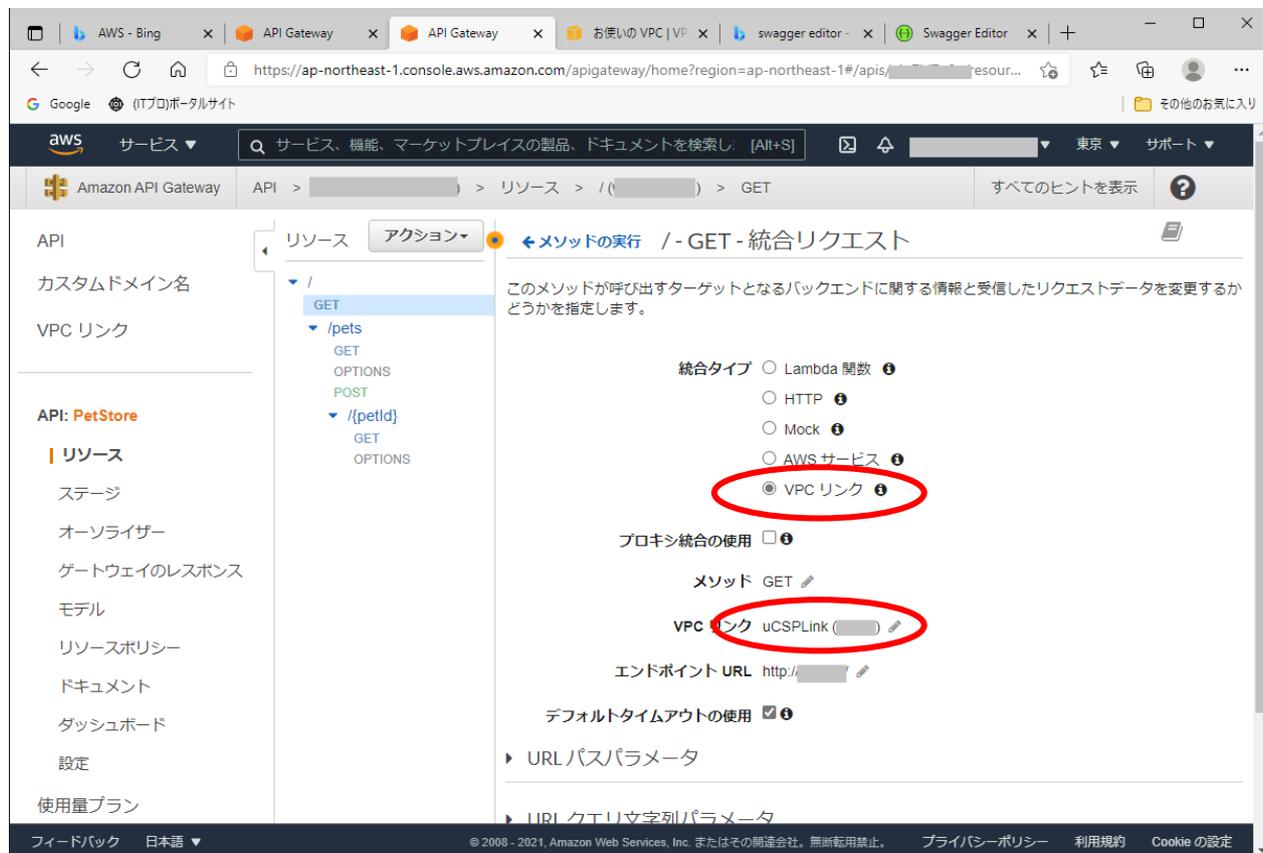
## 2.2.3 統合リクエストの設定

作成した VPC リンクを經由して、バックエンドの API サービスに接続するために、統合リクエストの設定をします。

API Gateway の画面の左メニューから、API の「リソース」を選択します。その後、「統合リクエスト」を選択します。



統合リクエストの設定では、統合リクエストの統合タイプとして「VPC リンク」を選択し、作成した VPC リンクを指定します。

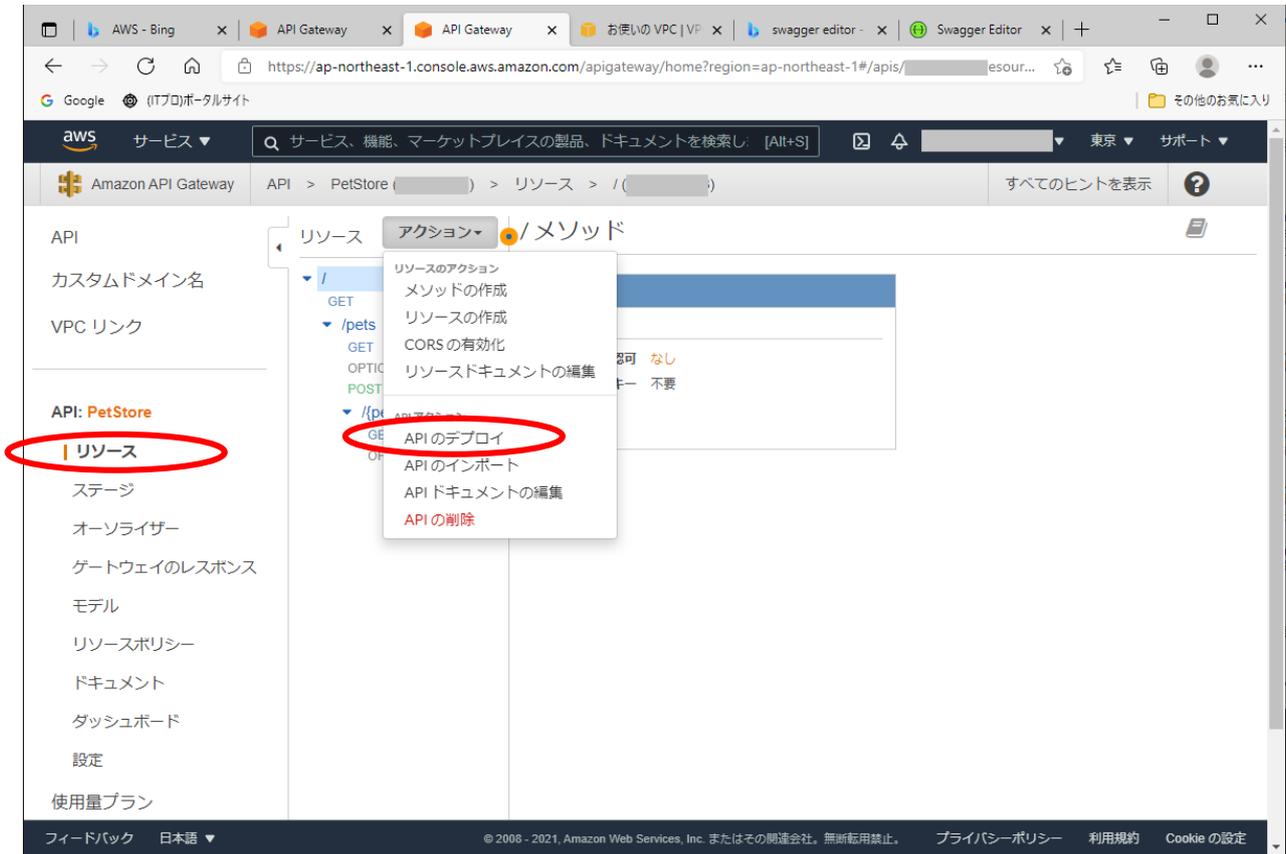


HTTP メソッドタイプや URL が、API Gateway のインターフェイスとバックエンドのサービスのインターフェイスで同一の場合、プロキシ統合を使用できます。プロキシ統合を使用しない場合は、API ごとにメソッドや URL を変更できます。

## 2.2.4 ステージの設定

作成した API を使用するためには、API をデプロイする必要があります。デプロイは、ステージと関連付けられません。

作成した API を選択し、左メニューから、「リソース」を選択します。その後、「アクション」を選択すると表示されるプルダウンから、「API のデプロイ」を選択します。



初回デプロイ時には、新しいステージを作成する必要があります。2 回目以降のデプロイ時には、既存のステージを上書きするか、新しいステージを作成することができます。

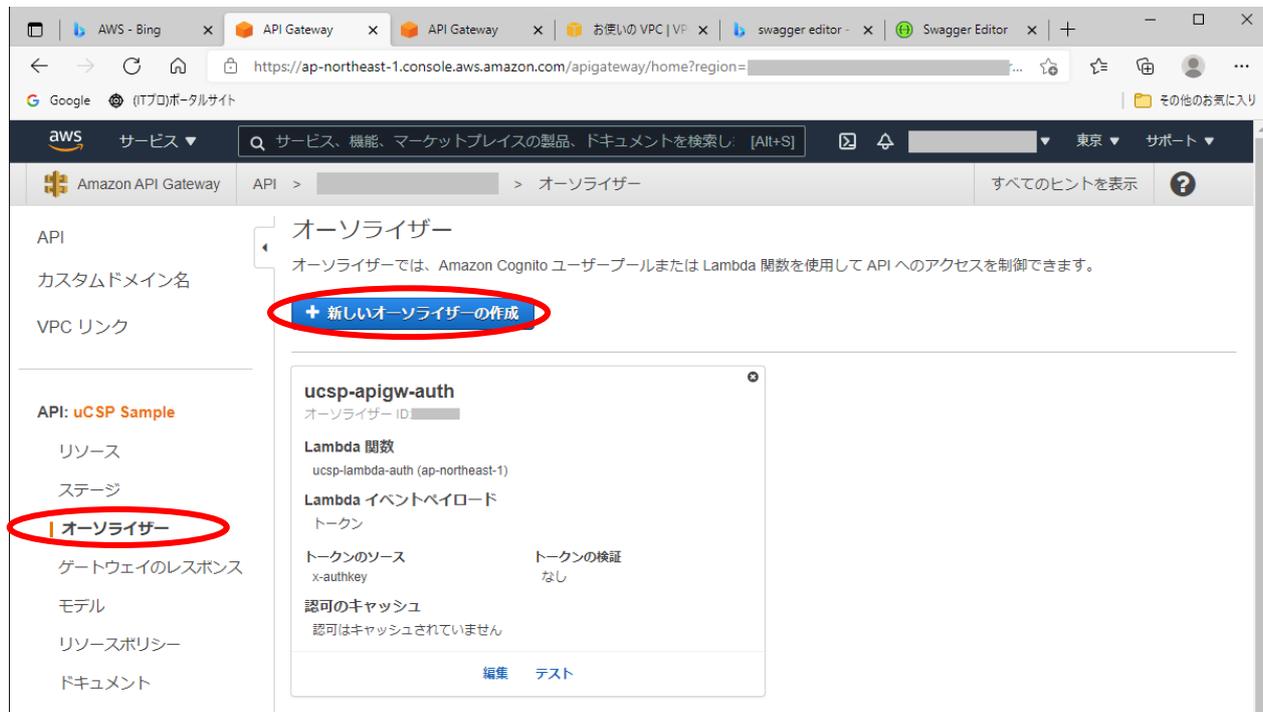
API Gateway の設定を変更した場合、再デプロイをする必要があります。

以上で EC2 上の REST API クライアントプログラムから、アクセス用インターフェイス VPC エンドポイントを経由して、作成した API Gateway を呼び出し、API Gateway から VPC リンク、Network Load Balancer を経由してバックエンドの uCosminexus Service Platform の REST API を呼び出すことができますようになります。

## 2.2.5 オーソライザーの設定

API Gateway のオーソライザーを設定することで、認証機能を使うことができます。認証機能は Lambda 関数で作成します。

API Gateway の画面の左メニューから、「オーソライザー」を選択し、「新しいオーソライザーの作成」から Lambda 関数を作成します。



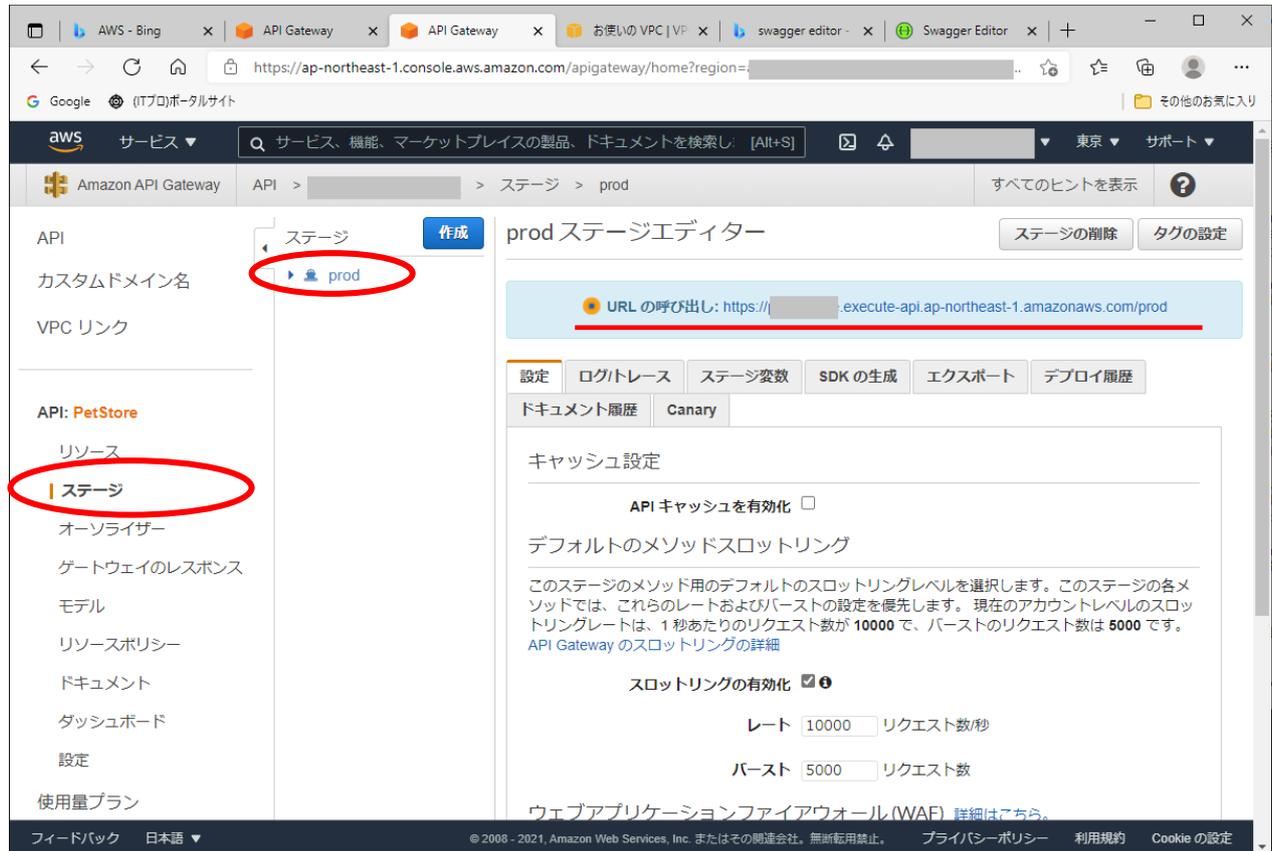
具体的な Lambda 関数の実装については、このドキュメントの対象外とします。

## 2.3 API Gateway の利用

### 2.3.1 API Gateway の呼び出し

API Gateway を呼び出すための URL は、ステージエディタで確認できます。

API Gateway の画面の左メニューから、「ステージ」を選択します。

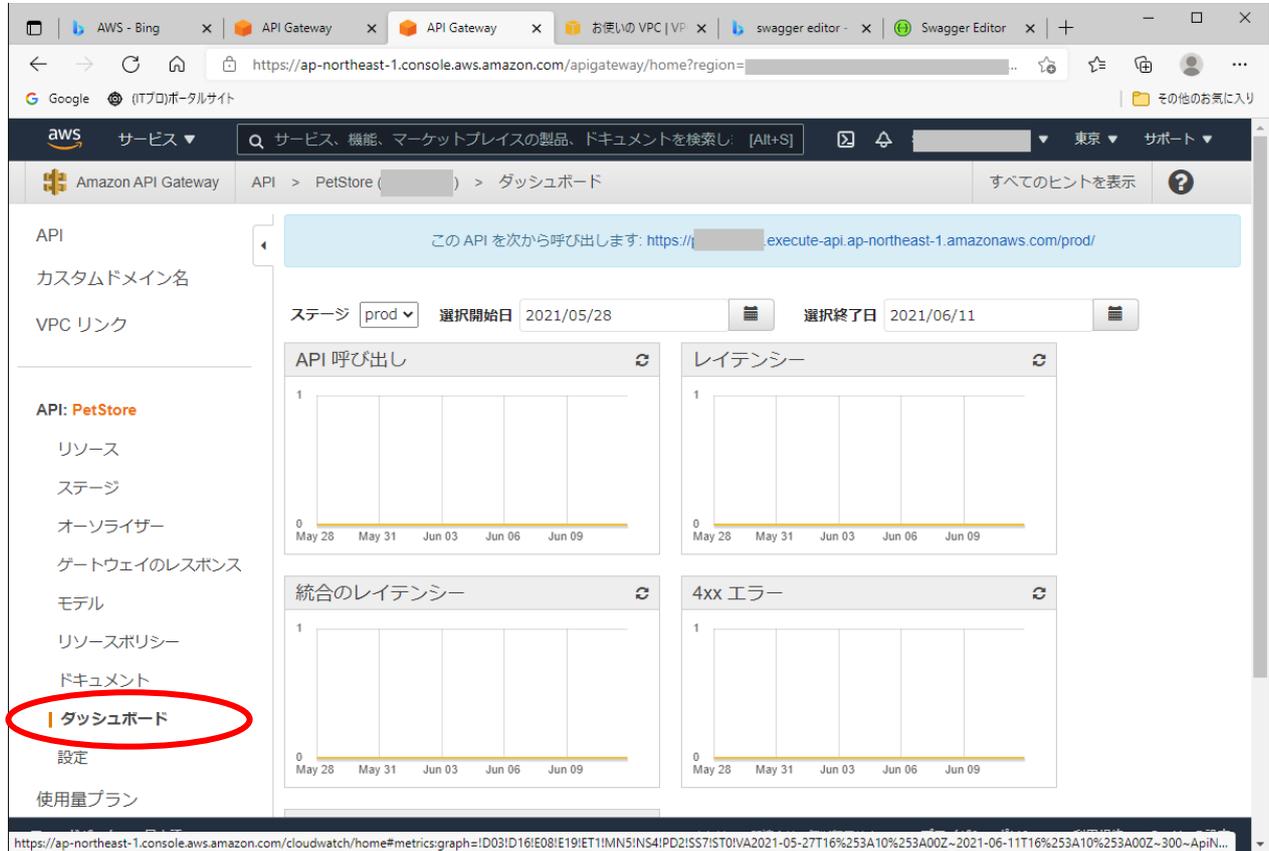


呼び出すステージを選択すると、URL が表示されます。URL のホスト名は AWS の提供するプライベート DNS で解決されます。

## 2.3.2 ダッシュボードの利用

API Gateway の呼び出しに関する統計情報は、ダッシュボードから参照できます。

API Gateway の画面の左メニューから、「ダッシュボード」を選択します。



The screenshot shows the AWS API Gateway console interface. The left-hand navigation menu is visible, with the 'ダッシュボード' (Dashboard) option highlighted with a red circle. The main content area displays the dashboard for the 'PetStore' API, showing the 'prod' stage. It includes a URL for the API, a date range selector (from 2021/05/28 to 2021/06/11), and four performance metrics: 'API 呼び出し' (API Calls), 'レイテンシー' (Latency), '統合のレイテンシー' (Integrated Latency), and '4xx エラー' (4xx Errors). Each metric is represented by a line graph with a y-axis from 0 to 1 and an x-axis showing dates from May 28 to June 9. The graphs currently show zero activity.

—以上—