

課題演習 No.2
(WAF+CloudFront+S3+動的コンテンツ)



課題演習2(WAF+CloudFront+S3+動的コンテンツ)

目的:

ある会社では世界中のユーザーに対してweb翻訳サイトをリリースしようと計画中です。アクセスは世界中から来るので、遅延なく低レイテンシーかつセキュアな配信したいと考えております。またAWSを使用した構築環境では、webの静的サイトと動的サイト(アプリケーション部分)の両方をサーバレスで実現したいとも考えています。翻訳のためのアプリケーションは開発部署で作成されたものを使用し、アプリケーションへの悪意ある攻撃に対してはセキュリティ保護を実装していくたいと考えています。

1:要件

■以下の要件でインフラを構築してください。

1. 世界中のユーザーへ低レイテンシーな配信環境で最高のエクスペリエンスを提供できるようにしたい。
2. 1の環境へのアクセスは HTTPSになるよう、セキュアなリクエスト受信をできるようにしたい。
3. ユーザーに表示させる静的コンテンツをマネージドサービスを利用したサーバレス環境で展開したい。
またセキュリティを考慮してオリジンページは世界に公開されないようにしたい。
4. 1で作成した環境に動的コンテンツも設定し、APIを使用したアプリケーションと連動できるようにする。
アプリケーションは開発部署で作成されたコードを使用して動作するようにする。
(**ヒント:アプリの権限(ロール)にTranslateReadOnlyを付与する**)
5. アプリケーションを対象にした悪意のある攻撃から保護できる環境にしたい。
(※実際に攻撃をして検証しなくてOKです)
6. 上記の環境で構築し、webページの閲覧とアプリケーションが動作し、翻訳コンテンツが問題無く動くことを確認する。

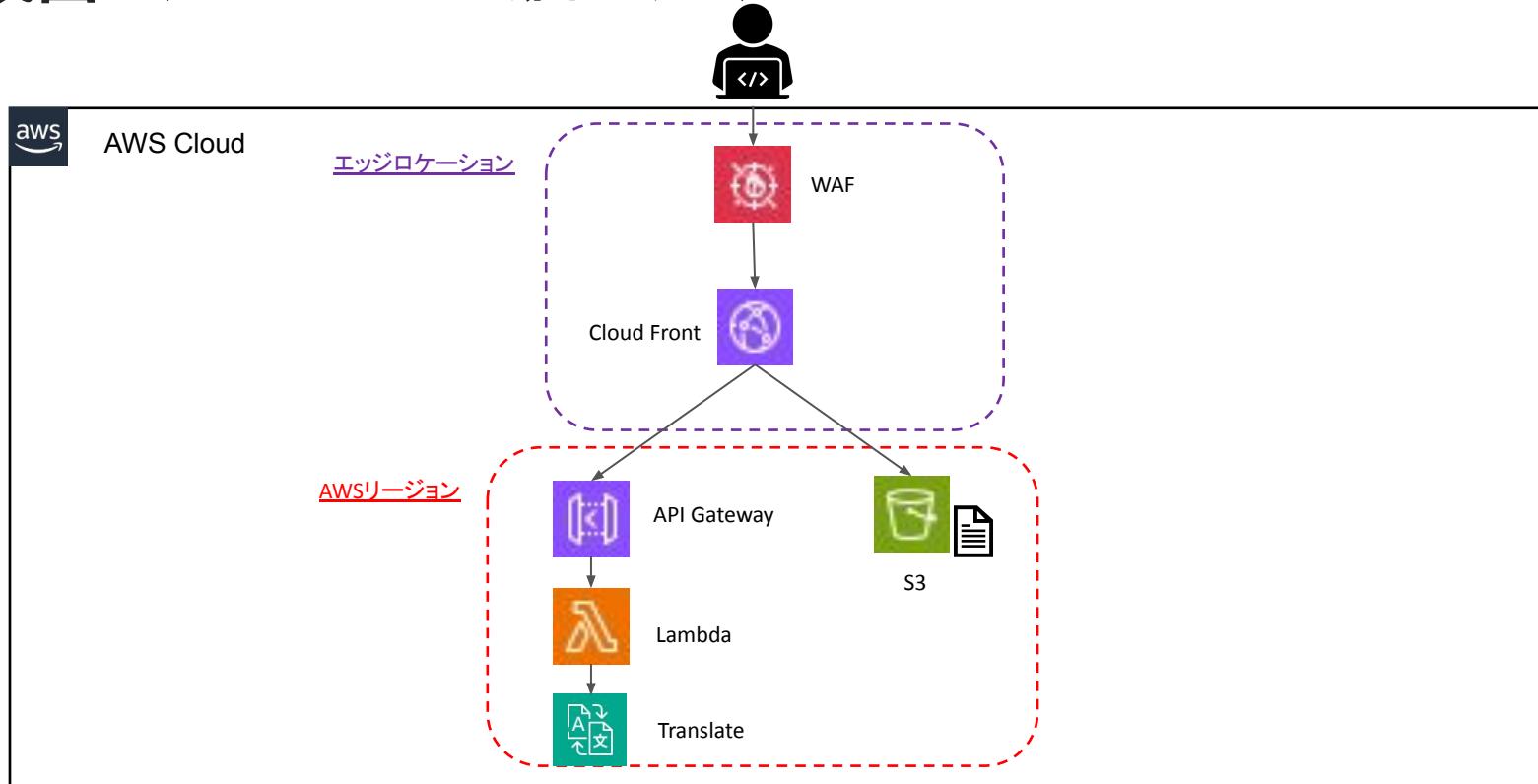
2: 設計

■要件に対しての仕様(解決策)

| No | 要件 | 仕様(解決策) |
|----|--|--|
| 1 | 世界中のユーザーへ低レイテンシーな配信環境で最高のエクスペリエンスを提供できるようにしたい。 | CloudFrontの設定をして、世界中のエッジロケーションからコンテンツを配信。 |
| 2 | ユーザーに表示させる静的コンテンツをマネージドサービスを利用したサーバレス環境で展開したい。またセキュリティを考慮してオリジンページは世界に公開されないようにしたい。 | CloudFront経由でHTTPSリクエストのみ許可。CloudFrontのオリジンにS3バケットを配置してバケット内に静的コンテンツ格納。OACを設定して、CloudFront経由のみS3へアクセス可能にする。S3バケットはパブリックアクセス不可。 |
| 3 | 1で作成した環境に動的コンテンツも設定し、APIを使用したアプリケーションと連動できるようにする。アプリケーションは開発部署で作成されたコードを使用して動作するようにする。 | CloudFrontのオリジンにAPI Gatewayを構築。API Gatewayの背後にLambdaを設定。アプリのロールにTranslateReadOnlyを付与。 |
| 5 | アプリケーションを対象にした悪意のある攻撃から保護できる環境にしたい。 | CloudFront設定時にWAFを同時に設定。 |
| 6 | 上記の環境で構築し、webページの閲覧とアプリケーションが動作し、翻訳コンテンツが問題無く動くことを確認する。 | WebブラウザでHTTPSアクセスし、静的コンテンツが正常に表示されるか確認。API Gatewayのエンドポイントへアプリからリクエストを送信し、期待通りのレスポンスが得られるか確認。 |

2: 設計

-構成図- (WAF+CloudFront+S3+動的コンテンツ)



3: 構築(実装)

① S3をパブリックアクセスを全てブロックしてセキュアに作成
作成したバケット内に静的コンテンツを格納
※CloudFrontのオリジンとして作成

Amazon S3 > バケット > バケットを作成

バケットは S3 に保存されたデータのためのコンテナです。

一般的な設定

AWS リージョン
アジアパシフィック(東京) ap-northeast-1

バケットタイプ | 情報

汎用
ほとんどのユースケースとアクセスパターンに推奨されます。汎用バケットは、これにより、複数のアベイラビリティーゾーンにまたがるオブジェクトクラスを組み合わせることができます。

バケット名 | 情報
test-bucket-85

バケット名はグローバル名空間内で一意であることに加えて、バケット命名規則

既存のバケットから設定をコピー - オプション
次の設定のバケット設定のみがコピーされます。

バケットを選択する

このバケットのブロックパブリックアクセス設定

パブリックアクセスは、アクセスコントロールリスト (ACL、Access Control List) を介して制御されます。このバケットとそのオブジェクトへの公開アクセスが確実にプロテクトされるようにするためにのみ適用されます。AWS では、パブリックアクセスをすべてブロックすることを確認ください。このバケットやオブジェクトへのアクセスを制御するには、ACL を使用してください。

パブリックアクセスをすべてブロック
この設定をオンにすることは、以下の 4 つの設定をすべてオフにすることと同じです。

新しいアクセスコントロールリスト (ACL) を介して付与された S3 は、新しく追加されたバケットまたはオブジェクトに適用された設定では、ACL を使用して S3 リソースへのパブリックアクセスを許可する新しい ACL を作成します。

任意のアクセスコントロールリスト (ACL) を介して付与された S3 は、バケットとオブジェクトへのパブリックアクセスを付与するすべての ACL を削除します。

新しいパブリックバケットポリシーまたはアクセスポイントポリシー S3 は、バケットとオブジェクトへのパブリックアクセスを許可する新しいポリシーを作成します。

任意のパブリックバケットポリシーまたはアクセスポイントポリシー S3 は、バケットとオブジェクトへのパブリックアクセスを付与するポリシーを削除します。

test-bucket-85 情報

オブジェクト | プロパティ | アクセス許可 | メトリクス

オブジェクト (2)

S3 URI をコピー URL をコピー ダウンロード

オブジェクトは、Amazon S3 に保存された基本的なエンティティです。オブジェクトにアクセスできるためには、明示的にアクセス権限を付与する必要があります。

検索: プレフィックスでオブジェクトを検索

| 名前 | タイプ |
|------------|------|
| index.html | html |
| static/ | フォルダ |

3: 構築(実装)

② 翻訳コンテンツを作成のためLambdaを起動

The screenshot shows the AWS Lambda function creation interface. At the top, there's a navigation bar with the AWS logo, search bar, and various service links like IAM, EC2, VPC, CloudFront, S3, Lambda, API Gateway, CloudFormation, Lightsail, WAF & Shield, Route 53, and AWS Application Migration Service. The region is set to アジアパシフィック(東京) (Asia Pacific (Tokyo)). Below the navigation bar, the path is Lambda > 関数 > 関数の作成.

Three options are available for creating a new function:

- 一から作成: シンプルな Hello World の例で開始します。
- 設計図の使用: 一般的なユースケース用のサンプルコードと設定プリセットから Lambda アプリケーションを構築します。
- コンテナイメージ: 関数にデプロイするコンテナイメージを選択します。

基本的な情報

関数名
関数の目的を名前として入力します。
test_lambda_kadai

関数名は 1 ~ 64 文字で、リージョン内で一意である必要があります。スペースを含めることはできません。有効な文字は a ~ z, A ~ Z, 0 ~ 9, ハイフン (-), およびアンダースコア (_) です。

ランタイム [情報](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13 ▼

アーキテクチャ [情報](#)
関数コードに必要な命令セットアーキテクチャを選択します。
 x86_64
 arm64

3: 構築(実装)

② 翻訳コンテンツ用のPythonコードをDeploy

```
lambda_function.py
import boto3

def lambda_handler(event, context):
    translate = boto3.client(service_name='translate', use_ssl=True)
    result = translate.translate_text(Text=event['queryStringParameters']['input_text'], SourceLang
    return {
        'statusCode': 200,
        'body': result
    }
```

3: 構築(実装)

②作成したLambdaのIAMロールにTranslateReadOnlyのポリシーをアタッチ

≡ IAM > ロール > [test_lambda_kadai2-role-lr3cqvni](#) > 許可を追加

ポリシーを test_lambda_kadai2-role-lr3cqvni にアタッチ

▶ 現在の許可ポリシー (1)

その他の許可ポリシー (1/1090)

The screenshot shows the AWS IAM Policy Editor interface. At the top, there's a breadcrumb navigation: '≡ IAM > ロール > [Role Name] > 許可を追加'. Below it, a heading says 'ポリシーを [Role Name] にアタッチ'. Underneath, a section titled '▶ 現在の許可ポリシー (1)' lists one policy. Further down, another section titled 'その他の許可ポリシー (1/1090)' shows a search bar with 'Transla' and a list of policies. The 'TranslateReadOnly' policy is highlighted with a blue border and checked, indicating it is selected.

| ポリシー名 | 状態 |
|---------------------|-------------------------------------|
| TranslateFullAccess | <input type="checkbox"/> |
| TranslateReadOnly | <input checked="" type="checkbox"/> |

3: 構築(実装)

③APIGatewayをRest APIでメソッドタイプGET、Lambda関数、プロキシ統合でアプリのLambdaを指定して作成

API Gateway > API > リソース - translational-api (bi1vxiyab1)

メソッドタイプ
GET

統合タイプ
 Lambda 関数
API を Lambda 関数と統合します。


AWS のサービス
AWS のサービスと統合します。


Lambda プロキシ統合
リクエストを構造化されたイベントとして Lambda 関数に送信します。

REST API を作成 情報

API の詳細

新しい API
新しい REST API を作成します。

API をインポート
OpenAPI 定義から API をインポートします。

API 名
translational-api

説明 - オプション

Lambda プロキシ統合
リクエストを構造化されたイベントとして Lambda 関数に送信します。

lambda 関数
lambda 関数の名前またはエイリアスを指定します。別のアカウントからの ARN を指定することもできます。

ap-northeast-1 ▾

Lambda 関数を呼び出すための許可を API Gateway に付与します。オフにするには、関数のリソースポリシーを自分自ら呼び出しロールを指定します。

応答のタイムアウト | 情報
デフォルトでは、50 ~ 29,000 ミリ秒のインテグレーションタイムアウトを入力できます。サービスクォータを使用して、統合タイムアウトを 29,000

3: 構築(実装)

- ③ APIGatewayの作成時の補足でクエリ文字列をinput_textで指定
ステージ名をapiで作成

29000

▶ メソッドリクエストの設定

▼ URL クエリ文字列パラメータ

名前 必須

[クエリ文字列を追加](#)

▶ HTTP リクエストヘッダー

...

ステージを作成

ステージの詳細

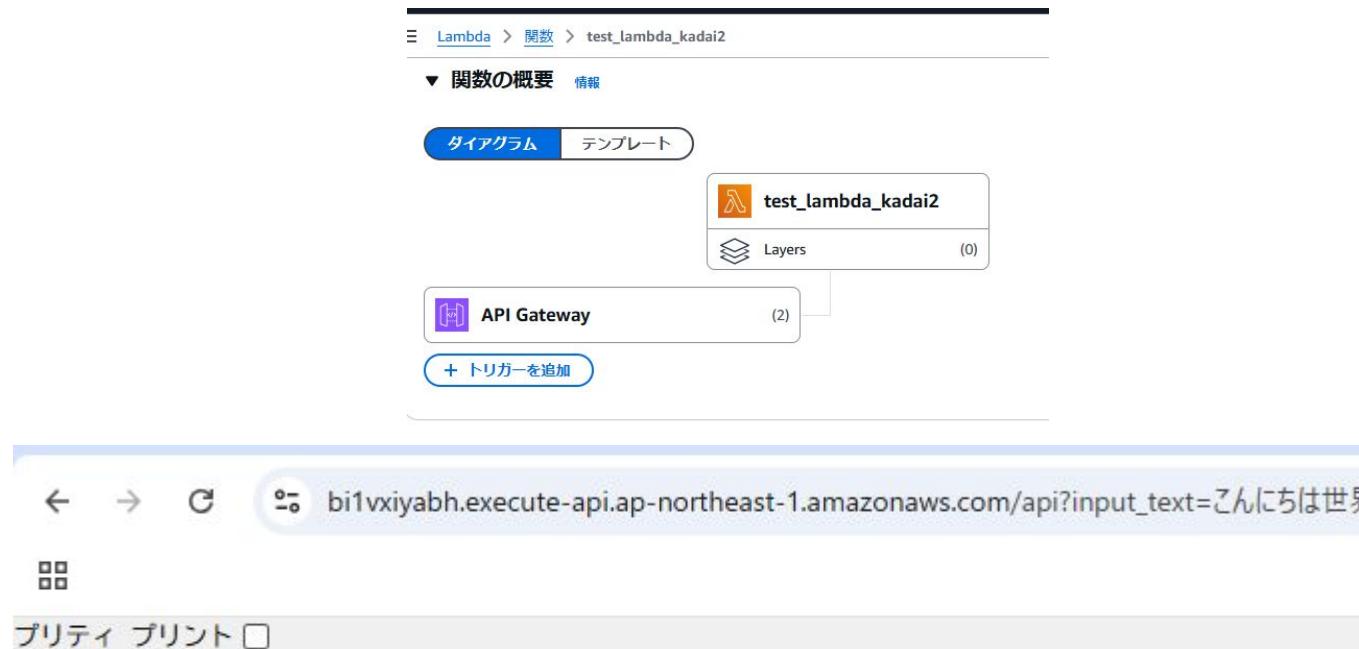
ステージ名

ステージの説明 - オプション

デプロイ

3: 構築(実装)

③ APIGateway作成時に作成したLambdaを指定し、Lambdaのトリガーとして設定



Hello world

3: 構築(実装)

④CloudFrontのディストリビューションを作成してデフォルトでS3バケットを指定
バケットへのアクセスをCloudFront経由のみ限定するためにOACを設定

ディストリビューションを作成

オリジン

Origin domain

Choose an AWS origin, or enter your origin's domain name. [Learn more](#)

test-bucket-85.s3.ap-northeast-1.amazonaws.com

Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC endpoint.

Origin path - optional

Enter a URL path to append to the origin domain name for origin requests.

Enter the origin path

名前

このオリジンの名前を入力します。

test-bucket-85.s3.ap-northeast-1.amazonaws.com

test-bucket-85.s3.ap-northeast-1.amazonaws.com

オリジンアクセス

情報

Public

Bucket must allow public access.

Origin access control settings (recommended)

Bucket can restrict access to only CloudFront.

Legacy access identities

Use a CloudFront origin access identity (OAI) to access the S3 bucket.

Origin access control

Select an existing origin access control (recommended) or create a new one.

test-bucket-85.s3.ap-northeast-1.amazonaws.com

フルサムネルマーク (マーク) を使用する CloudFront バケット

キャッシュキーとオリジンリクエスト

キャッシュキーとオリジンリクエストを制御するには、キャッシュ

Cache policy and origin request policy (recommended)

Legacy cache settings

キャッシュポリシー

既存のキャッシュポリシーを選択するか、新しいキャッシュポリシ

Caching Optimized

Policy with caching enabled. Supports Gzip and Brotli compressio

[Create cache policy](#) ポリシーを表示

オリジンリクエストポリシー - オプション

既存のオリジンリクエストポリシーを選択するか、新しいオリジン

[オリジンポリシーを選択](#)

[Create origin request policy](#)

3: 構築(実装)

④CloudFrontのディストリビューションを作成してデフォルトでS3バケットを指定
バケットへのアクセスをCloudFront経由のみ限定するためにOACを設定

ディストリビューションを作成

オリジン

Origin domain

Choose an AWS origin, or enter your origin's domain name. [Learn more](#)

test-bucket-85.s3.ap-northeast-1.amazonaws.com

Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC endpoint.

Origin path - optional

Enter a URL path to append to the origin domain name for origin requests.

Enter the origin path

名前

このオリジンの名前を入力します。

test-bucket-85.s3.ap-northeast-1.amazonaws.com

test-bucket-85.s3.ap-northeast-1.amazonaws.com

オリジンアクセス

情報

Public

Bucket must allow public access.

Origin access control settings (recommended)

Bucket can restrict access to only CloudFront.

Legacy access identities

Use a CloudFront origin access identity (OAI) to access the S3 bucket.

Origin access control

Select an existing origin access control (recommended) or create a new one.

test-bucket-85.s3.ap-northeast-1.amazonaws.com

フルサムネルマーク (マーク) を使用する CloudFront バケット

キャッシュキーとオリジンリクエスト

キャッシュキーとオリジンリクエストを制御するには、キャッシュ

Cache policy and origin request policy (recommended)

Legacy cache settings

キャッシュポリシー

既存のキャッシュポリシーを選択するか、新しいキャッシュポリシ

CachingOptimized

Policy with caching enabled. Supports Gzip and Brotli compressio

[Create cache policy](#) ポリシーを表示

オリジンリクエストポリシー - オプション

既存のオリジンリクエストポリシーを選択するか、新しいオリジン

[オリジンポリシーを選択](#)

[Create origin request policy](#)

3: 構築(実装)

④作成したディストリビューションからAPIGatewayもオリジンにするためオリジンを作成
APIGatewayのビヘイビアを作成

CloudFront > ティストリビューション > EFGKN4K72BADH >
オリジンを作成

設定

Origin domain

Choose an AWS origin, or enter your origin's domain name. [Learn more](#)

bi1vxiyah.execute-api.ap-northeast-1.amazonaws.com

Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC origin

プロトコル

- HTTPのみ
- HTTPSのみ
- マッチピューワー

HTTPS port

Enter your origin's HTTPS port. The default is port 443.

443

ビヘイビアを作成

設定

パスパターン | [情報](#)

api

オリジンとオリジングループ

bi1vxiyah.execute-api.ap-northeast-1.amazonaws.com

キャッシングキーとオリジンリクエストを制御するには、キ...

- Cache policy and origin request policy (recommended)
- Legacy cache settings

キャッシングポリシー

既存のキャッシングポリシーを選択するか、新しいキャッシングポリ...

CachingDisabled

Policy with caching disabled

[Create cache policy](#) [ポリシーを表示](#)

オリジンリクエストポリシー - オプション

既存のオリジンリクエストポリシーを選択するか、新しいオリジ...

input_text2

[Create origin request policy](#) [ポリシーを表示](#)

3: 構築(実装)

⑤悪意のある攻撃から保護する為に、CloudFront作成時にWAFを設定

The screenshot shows the AWS CloudFront distribution creation process. The current step is 'ウェブアプリケーションファイアウォール (WAF) 情報' (Web Application Firewall Information). A callout box highlights the 'セキュリティ保護を有効にする' (Enable security protection) option, which uses AWS WAF to protect against common web threats. Below this, there is a 'Use monitor mode' checkbox with explanatory text.

セキュリティ保護を有効にする
AWS WAF を使用して、最も一般的なウェブの脅威やセキュリティの脆弱性からアプリケーションを保護します。ブロックされたリクエストは、ウェブサーバーに到達する前に停止されます。

Use monitor mode
Count how many of your requests would be blocked by this WAF configuration. When ready, you can disable requests.

4: テスト結果(検証)

- ① S3バケットへの直接アクセスができないように設定されているか
確認のためバケット内のコンテンツURLをブラウザ検索
拒否されたためバケットへの制限がされているのを確認

The screenshot shows a web browser window with the URL `test-bucket-85.s3.ap-northeast-1.amazonaws.com/index.html`. The page content is as follows:

```
This XML file does not appear to have any style information associated with it. The document tree is shown below.

▼<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>C6Q2G2VPXX0865QD</RequestId>
  <HostId>3z0FPQ6q78drMB4iFW9schk4DHqwR7m4HKvPIC5TrAqj3k8jG00KmPyRjQI2B41X7mgjJ+SfIJ4=</HostId>
</Error>
```

4: テスト結果(検証)

②CloudFrontのディストリビューションドメインをブラウザから
アクセスをし、無事に翻訳機能が動作する事を確認

The screenshot shows a web browser window with the following details:

- Address Bar:** https://durlzd96ky0x5.cloudfront.net
- Toolbar:** Google Lens, Refresh button, and a link to 'すべてのブックマーク'.
- Content Area:**
 - A purple circular icon with a globe and lines.
 - Input Message:** A text input field containing "こんにちは世界".
 - Buttons:** Send and All Clear.
 - Output Message:** A text input field containing "Hello world".