復習課題(EC2/Dynamo)演習2-1

クラス名　　　　No. 氏名

今回の演習では、Lambda/APIGateway、DynamoDBを利用したWebシステムを構築します。EC2上のWebサーバからAPIを利用してDynamoDBの内容を表示します。

※Leaner Labで作成すること。

※設定情報は「Webサイト設定シート」ファイルに入力していくこと。

ダイアグラム

自動的に生成された説明

■キーペアの作成

1.キーペアがない場合、クラス名-key.pemで作成しておくこと。

　※すでに使用できるキーペアがあれば、そちらを使用すること。

■ネットワークの構築

VPC、サブネットなどネットワークの構築を行いなさい。

1.VPCを作成しなさい。

　VPC名：ecc\_vpc CIDR: 10.2.0.0/16

2.次のサブネットを作成しなさい(VPC：ecc\_vpc)。

①パブリックサブネット用・・・サブネット名：ecc\_pub\_sub　AZ：us-east-1a　CIDR：10.2.1.0/24

3.インターネットゲートウェイを作成しなさい。

　名前：ecc\_igw　アタッチ：ecc\_vpc

4.次のルートテーブルを作成しなさい。＊VPCはecc\_vpc

　①ecc\_pub\_sub用・・・ルートテーブル名：ecc\_rt

5.次のようにサブネットのルートテーブルの関連付けを行いなさい。

　①ecc\_rt・・・ecc\_pub\_sub

6.次のセキュリティグループを作成しなさい。＊設定はすべて「インバウンドルール」で行うこと。

　①EC2用(ecc\_websv)

グループ名：ecc\_websv\_sg　説明：security group for ecc\_websv 　VPC：ecc\_vpc

タイプ：SSH ソース：0.0.0.0、タイプ：HTTP ソース：0.0.0.0)

7.ルートテーブル(ecc\_rt)に次のルートを追加しなさい。

　送信先：0.0.0.0 ターゲット：インターネットゲートウェイ(ecc\_igw)

■EC２の設定(Webサーバ)

1.次の仕様でインスタンスを起動しなさい。

・キーペア　・・・　クラス名-key.pem　※使用可能なキーがあれば作成しなくてもよい。

・VPC：ecc\_vpc　サブネット：ecc\_pub\_sub 名前：ecc\_websv インスタンスタイプ：t2.micro

パブリックIPの自動割り当て：有効化　 セキュリティグループ：ecc\_websv\_sg

■接続確認/設定

1.Tera termでWebサーバ(ecc\_websv)に接続しなさい。＊パブリックアドレスでアクセスする

2.Webサーバ(Apache)とPHPをインストールしなさい。

|  |
| --- |
| sudo yum install -y httpd  sudo yum install -y php php-fpm php-mysqli php-json php-devel  sudo systemctl start httpd  sudo systemctl enable httpd |

3.次の内容のPHPファイルを作成して、アップロードしてPHPの動作確認をしなさい。/var/www/htmlへ移動する。

|  |
| --- |
| <?php  phpinfo();  ?> |

■DynamoDBの設定

1.テーブルの作成(DynamoDB＞テーブル)

　次の仕様に従いテーブルを作成しなさい。

　テーブル名：ecc-cake-table　パーティションキー：pid(数値)

タグ：キー(Name)、値(ecc-cake-table)

＊状態が「Active」になったらOK。

＊テーブルがすでになる場合は、内容の変更のみでもOK。

2.データの追加

　「項目を表示」を選び、「項目の作成」からデータを追加しなさい。

＊項目のタイプ・・・pname(文字列)、category(文字列)、photo(文字列)、price(数値)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pid | pname | category | Photo | price |
| 1001 | ブッシュド・ノエル | ショートケーキ | cake01.jpg | 250 |
| 1002 | シブースド・ノエル | ショートケーキ | cake02.jpg | 200 |
| 1003 | イチゴとシブースドケーキ | ホール | cake03.jpg | 2000 |

復習課題(Lambda/APIGateway)演習2-2

クラス名　　　　No. 氏名

■Lambda関数の作成

1.関数を作成しなさい。詳細は以下のとおり。

　一から作成　関数名：ecc-cake-function　ランタイム：Python3.13

　実行ロール：既存のロールを使用する(LaboRole)

2.コードの入力

lambda\_functionの内容をすべて削除して、次のコードをコピーしなさい。(lambda-func01.txt)。

|  |
| --- |
| import boto3  from boto3.dynamodb.conditions import Key, Attr    # DynamoDBと接続  dynamodb = boto3.resource('dynamodb')    # テーブル内のIDを指定してデータを該当するitemを取得する  def lambda\_handler(event, context):  table\_name = "クラス名番号-cake-table"  partition\_key = {"pid": event["pid"]}  dynamotable = dynamodb.Table(table\_name)  res = dynamotable.get\_item(Key=partition\_key)  item = res["Item"]    return item |

3.作成した関数をデプロイしなさい。

4.作成した関数を実行する。

　今回はトリガーなしで関数を実行する。まずは「Test」をクリックしてテストイベントの設定を行いなさい。

　イベント名：ecc-event１

　「テストイベント」に次のコードを入れる(いらないものを削除して修正を行う)。

|  |
| --- |
| {  "pid": 1001  } |

5.関数を実行しなさい。

6.次の内容が表示されるか確認しなさい。

|  |
| --- |
| Response  {  "pid": 1001,  "category": "ショートケーキ",  "pname": "ブッシュド・ノエル",  "photo": "cake01.jpg",  "price": 250  } |

■トリガーの作成(Lambda>関数)

1.Lambda関数(ecc-cake-function)を選び、「トリガーを追加」をクリックする。ソースの選択から「API Gateway」を選択しなさい。

2.以下の使用でAPIを作成しなさい(「新規APIを作成」を選ぶ)。

　　APIタイプ：REST API セキュリティ：IAM　　API名：ecc-cake-function-API

3.「トリガー」に追加された確認しなさい。

■API Gatewayの作成(API Gatewayで実施)

1.作成したAPI GatewayをクリックしてAPI Gatewayに移動。メソッドの「ANY」を選択して、「削除」しなさい。

2.クラス名番号-cake-functionを選び、「メソッドを作成」からメソッドを作成しなさい。

　メソッドのタイプ：GET　　Lambda関数：クラス名番号-cake-function

　GETメソッドは作成できたか確認しなさい。[　確認できた　]

3.API Gatewayのテスト次の仕様に従い行いなさい。

　①「統合リクエスト」タブを選択して、次の設定をおこないなさい。ここでは、実際に送信するデータなどの指定を行う。

　　　・マッピングテンプレートを選び、「マッピングテンプレートの追加」をクリック

　　　　コンテンツタイプ：application/json テンプレート：Empty

②テンプレート欄に次のコードを入力しなさい。入力後に「保存」しなさい。

|  |
| --- |
| {  "pid" : $input.params('pid')  } |

　　　　※URLで受け取ったパラメータを取得する。

③「テスト」タブを選択して次の設定を行いなさい。

　　　・クエリ文字列・・・pid=1001

入力後に「テスト」ボタンをクリックして、テストしなさい。

④レスポンス本文に次のような表示ができれば、テストは成功。テストは成功したか。[　成功　]

|  |
| --- |
| レスポンス本文  {"pid": 1001, "pname": "ブッシュド・ノエル", "price": 200} |

　　　※成功した場合、API GatewayからLambda関数が実行されたことが確認できたことになる。

　⑤テストが終了したら、「APIのデプロイ」を選ぶ。「ステージ」を「default」に設定してデプロイしなさい。

復習課題(Lambda/APIGateway/S3)演習2-3

クラス名　　　　No. 氏名

■APIGatewayの確認(ブラウザで確認する)

1.Lambdaに戻り、「トリガー」の「API Gateway」を選び、「API endpoint」のURLをコピーする。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

2.ブラウザで1でコピーしたURLを貼り付け、?pid=1001を追加して実行しなさい。

　　＊https://**～?pid=1001**

3.次のような表示がされた確認しなさい。[　確認できた　]



■S3のバケット作成

１.次の仕様に従いバケットを作成しなさい。

　バケットのタイプ：汎用　バケット名：クラス名番号-cake-bucket リージョン：バージニア北部(us-east-1)

　ACL有効 バケットのバージョニング：有効にする　パブリックアクセスをすべてブロック：チェックを外す

2.1で作成したバケットにフォルダ「images」を作成し、cake01.jpg〜cake03.jpgをアップロードしなさい。

3.アップロードした3つのファイルとimagesフォルダを「ACLを使用して公開」を選び公開するように設定すること。

4.設定シートのS3の箇所に情報を記入しなさい。

5.作成したバケット内の「images」フォルダを選び、「URLをコピー」して一旦テキストファイルにペーストしておく。

　＊ここでコピーしたものは、画像表示の時にsrcに指定する。

■VPCエンドポイントの作成

プライベートサブネット内のEC2からS3のバケットにアクセスするためのエンドポイントを作成しなさい。

1.次の仕様に従いVPCエンドポイントを作成しなさい。

　名前タグ：クラス名番号-s3-endpoint サービスカテゴリ：AWSのサービス

　サービス名：com.amazonaws-us-east-1.s3(タイプ：Gateway) VPC：ecc\_vpc

ルートテーブル：ecc\_rt ポリシー：フルアクセス

■PHPファイルの変更

1.画像ファイルが表示されるようにPHPファイル(revi2.php)を修正しなさい。

|  |
| --- |
| <img src=エンドエンドポイント(コピーしたURL)/images/%s> |

　　　↓ コピーしたURLを設定

<設定例>

|  |
| --- |
| <img src=https://ie3a000-bucket.s3.amazonaws.com/images/%s> |

■PHPで作成したAPIからデータを取得して、実行する。

　＊ここでは以前に作成したEC2の「クラス名番号\_websv」にPHPファイルをアップロードして確認すること。

1.PHPファイル(revi2.php)を修正しなさい。「API GatewayのURL (endopoint)」の部分を自分のAPI

GatewayのURLに変更しなさい。

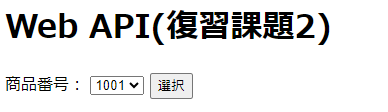
|  |
| --- |
| $url = 'API GatewayのURL(endopoint)?pid=' . $number; |

2.変更後、Tera TermでEC2にアクセスして、part-db2.phpをアップロードしなさい。

3.アップロード後、ファイルを「/var/www/html」へ移動しなさい。

4.ブラウザで次のURLにアクセスして次のような表示がされるか確認しなさい。

http://IPアドレス/revi2.php



＊商品番号を選択し、「選択」ボタンを

　　　クリックした結果

5.ワードファイルを「クラス名番号名前\_復習演習２」作成したPHPファイルを、「クラス名番号名前\_revi2.php」、設

定シートを「クラス名番号名前\_復習演習2設定シート」で保存してsv23に提出しなさい。