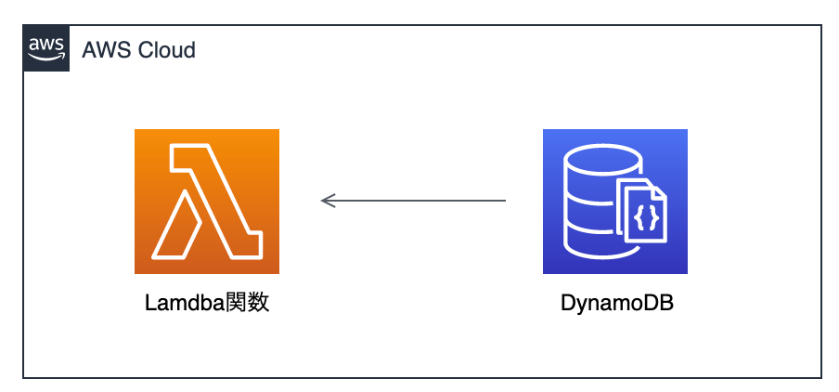
DynamoDB演習

クラス名　　　　No. 氏

今回の演習ではDynamoDBをLambda(関数)で操作します。DynamoDBはNoSQL(キー・バリュー型)のデータベースになります。※Leaner Labで実施してください。



■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：クラス名番号-cake-table　パーティションキー：pid (数値)

タグ：キー(Name)、値(クラス名番号-db)

＊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)、price(数値)

|  |  |  |  |
| --- | --- | --- | --- |
| pid | pname | category | price |
| 1 | ブッシュド・ノエル | ショートケーキ | 250 |
| 2 | イチゴとシブースドケーキ | ショートケーキ | 400 |
| 3 | クリスマスプディング | ホール | 2000 |

■Lambdaの設定1(データの抽出)

1.次の仕様に従い、Lambda関数(一から作成)を作成しなさい。

　関数名：cake-dyDB-func　ランタイム：Python 3.13

実行ロール：既存のロールを使用(LabRole)

　関数は作成でき方確認しなさい。［　確認できた　］

2.関数の内容を書き換えなさい。書き換える内容はcake-dyDB-func.txtを参照すること。

　＊NT22から「cake-dyDB-func.txt」をコピーしてテーブル名を修正しなさい。

　＊作成後に「Deploy」をしておくこと。

3.テストイベントの設定

　実行のテストするための準備を次のようなテストイベントを作成しなさい。イベント名：クラス名dyDB-event1

　設定後に「保存」、「テスト」を選び動作確認をしなさい。

　次のような結果(データがすべて)が表示されたか確認しなさい。[　確認できた　]

テキスト

自動的に生成された説明

■Lambdaの設定2(データの書き込み)

1.DynamoDBへ指定したデータを追加するために、次のコードを追加しなさい。追加する場所はコメントの「追加処理」」の下に記述する。

|  |
| --- |
| #データ取得  pid = event['pid']  pname = event['pname']  category = event['category']  price = event['price']  #追加データの作成  addtional\_item = {  'pid': pid,  'pname': pname,  'category': category,  'price': price  }  #テーブルへデータを追加  ret = table.put\_item(Item=addtional\_item) |

追記後に「Deploy」する。

25.テストのイベント内容を変更

　「テスト」タブを選び、「イベントJSON」の内容を次のように変更しなさい。

イベントの内容

|  |
| --- |
| {  "pid": 4,  "pname": "シブースド・ノエル",  "category": "ショートケーキ",  "price": 200  } |

6.保存後にテストを実行し、「Function Logs」の内容を確認しなさい。

　　イベントの内容に記述したデータは追加されたか。[　追加された　]

　　追加された部分の内容を記述しなさい。

|  |
| --- |
| Function Logs:  START RequestId: 81266a7a-58f0-4926-ba99-12bf68ba2c34 Version: $LATEST  name:クリスマスプディング categoryホール price:2000  name:イチゴとシブースドケーキ categoryショートケーキ price:400  name:シブースド・ノエル categoryショートケーキ price:200  name:ブッシュド・ノエル categoryショートケーキ price:250  END RequestId: 81266a7a-58f0-4926-ba99-12bf68ba2c34  REPORT RequestId: 81266a7a-58f0-4926-ba99-12bf68ba2c34  Duration: 643.26 ms Billed Duration: 644 ms Memory Size: 128 MB Max Memory Used: 78 MB  Request ID: 81266a7a-58f0-4926-ba99-12bf68ba2c34 |

■提出

　テストで使用したPythonのコードをコピーして「dynamoDB-番号名前.txt」で保存してsv23に提出しなさい。