Nhóm 1 đề tài 11: Xây dựng ứng dụng trên AWS cho phép tạo Database và cung cấp api để thêm, xóa, sửa trên Database

Thành viên tham gia Project:

+ Lý Quốc Dũng - 19133015 - 19133015@student.hcmute.edu.vn

+ Nguyễn Huỳnh Minh Trung - 19133061 - 19133061@student.hcmute.edu.vn

+ Bùi Thị Ngân Tuyền - 19133066 - 19133066@student.hcmute.edu.vn

**TÌM HIỂU VỀ API GATEWAY AWS**

**1. API là gì?**

- API là viết tắt của Application Programming Interface – phương thức trung gian kết nối các ứng dụng và thư viện khác nhau.

- Nó cung cấp khả năng truy xuất đến một tập các hàm hay dùng, từ đó có thể trao đổi dữ liệu giữa các ứng dụng

- Có 2 loại API AWS: API WEBSOCKET và API RESTful

**2. Ưu Điểm:**

2.1. Phát triển API hiệu quả

Chạy đồng thời nhiều phiên bản của cùng một API với API Gateway, từ đó cho phép bạn lặp lại, kiểm tra và phát hành các phiên bản mới một cách nhanh chóng. Bạn trả tiền cho các lệnh gọi được thực hiện với các API của mình và chuyển dữ liệu ra và không có lệ phí tối thiểu hay cam kết trả trước.

2.2. Hoạt động ở mọi quy mô

Cung cấp cho người dùng cuối độ trễ thấp nhất có thể cho các yêu cầu và phản hồi API bằng cách tận dụng mạng lưới các vị trí biên bằng cách sử dụng Amazon CloudFront. Điều tiết lưu lượng truy cập và cấp phép các cuộc gọi API nhằm đảm bảo rằng các hoạt động backend chịu được các lưu lượng truy cập tăng đột biến và các hệ thống backend không được gọi một cách không cần thiết.

2.3. Tiết kiệm chi phí ở quy mô lớn

API Gateway cung cấp mô hình giá theo bậc cho các yêu cầu API. Với mức giá API thấp 0.90 USD trên một triệu yêu cầu ở bậc cao nhất, bạn có thể giảm chi phí khi mức độ sử dụng API tăng lên trên mỗi khu vực trong toàn bộ các tài khoản AWS.

2.4. Theo dõi dễ dàng

Giám sát các số liệu về hiệu năng và thông tin về các lệnh gọi API, độ trễ dữ liệu và tỷ lệ lỗi từ bảng điều khiển API Gateway, nhờ đó bạn có thể giám sát trực quan các lệnh gọi đến dịch vụ của bạn thông qua Amazon CloudWatch.

2.5. Kiểm soát bảo mật linh hoạt

Cho phép truy cập các API của bạn bằng AWS Identity and Access Management (IAM) và Amazon Cognito. Nếu bạn sử dụng token OAuth, API Gateway có hỗ trợ OIDC gốc và OAuth2. Để hỗ trợ các yêu cầu cấp phép tùy chỉnh, bạn có thể thực thi trình cấp quyền Lambda từ AWS Lambda.

2.6. Tùy chọn API RESTful

Tạo API RESTful bằng API HTTP hoặc API REST. API HTTP là cách tốt nhất để xây dựng API cho hầu hết các trường hợp sử dụng. Các này rẻ hơn tới 71% so với API REST. Nếu cần chức năng proxy API và các tính năng quản lý trong cùng một giải pháp cho trường hợp sử dụng của mình, thì bạn có thể dùng API REST.

**3. API RESTful**

- API RESTful là một giao diện mà hai hệ thống máy tính sử dụng để trao đổi thông tin một cách an toàn qua internet. Hầu hết các ứng dụng kinh doanh phải giao tiếp với các ứng dụng nội bộ và bên thứ ba khác để thực hiện tác vụ.

- Các Method:

· POST – Create: Tạo dữ liệu mới

· GET – Read: Lấy dữ liệu về

· PUT – Update: Cập nhật dữ liệu

· DELETE – Delete: Xóa dữ liệu

**4. Hoạt động của API REST:**

B1. Client gửi một yêu cầu đến máy chủ. Client làm theo tài liệu API để định dạng yêu cầu theo cách mà máy chủ hiểu được.

B2. Máy chủ xác thực client và xác nhận rằng client có quyền đưa ra yêu cầu đó.

B3. Máy chủ nhận yêu cầu và xử lý trong nội bộ.

B4. Máy chủ trả về một phản hồi đến client. Phản hồi chứa thông tin cho client biết liệu yêu cầu có thành công hay không. Phản hồi cũng bao gồm bất kỳ thông tin nào mà client yêu cầu.

**5. Mô hình API REST:**

Diagram

Description automatically generated

**API Document Project:**

**Những API tạo bằng API Gateway**

**GetAllTableDetails**

https://aytm5kpem6.execute-api.us-east-1.amazonaws.com/gettabledetails/api/v1/gettabledetails

Phương thức: POST

Miêu tả: Trả về thông tin của các table có trong dynamodb

Tham số truyền vào: Không yêu cầu tham số

Ví dụ về kết quả trả về:

{

    "table1": {

        "PartitionKey": "key",

        "SortKey": "",

        "Status": "ACTIVE"

    },

    "table2": {

        "PartitionKey": "key",

        "SortKey": "",

        "Status": "ACTIVE"

    },

    "table3": {

        "PartitionKey": "key",

        "SortKey": "sortKey",

        "Status": "ACTIVE"

    },

    "table4": {

        "PartitionKey": "???",

        "SortKey": "????",

        "Status": "ACTIVE"

    },

    "table5": {

        "PartitionKey": "???",

        "SortKey": "????",

        "Status": "ACTIVE"

    }

}

**GetItemsByTableName**

https://2l8rbpvqqd.execute-api.us-east-1.amazonaws.com/getitemsbytablename/api/v1/getitemsbytablename

Phương thức: POST

Miêu tả: Trả về tất cả các Items cùng một số thông tin liên quan dựa trên tên bảng truyền vào

Ví dụ về tham số truyền vào:

{

  "TableName": "table1"

}

Ví dụ về kết quả trả về:

{

    "Items": [

        {

            "value": "item2",

            "key": "2",

            "CCCD": "191"

        },

        {

            "key": "1",

            "value": "item1"

        },

        {

            "key": "3",

            "value": "item3"

        }

    ],

    "Count": 3,

    "ScannedCount": 3,

    "ResponseMetadata": {

        "RequestId": "U0II3D3A6U9VS9MUOUF6HRRFHBVV4KQNSO5AEMVJF66Q9ASUAAJG",

        "HTTPStatusCode": 200,

        "HTTPHeaders": {

            "server": "Server",

            "date": "Sat, 28 May 2022 07:38:57 GMT",

            "content-type": "application/x-amz-json-1.0",

            "content-length": "177",

            "connection": "keep-alive",

            "x-amzn-requestid": "U0II3D3A6U9VS9MUOUF6HRRFHBVV4KQNSO5AEMVJF66Q9ASUAAJG",

            "x-amz-crc32": "3929514541"

        },

        "RetryAttempts": 0

    }}

**Những API tạo bằng SQS**

**CreateTable**

https://sqs.us-east-1.amazonaws.com/549480557245/TriggerCreateTable

Miêu tả: Tạo một Dynamodb Table dựa trên tham số truyền vào

Ví dụ về Message Body:

{

  "TableName": "Employees",

  "HashKeyName": "Name",

  "HashKeyType": "S",

  "RangeKeyName": "Email",

  "RangeKeyType": "S",

  "ReadCapacityUnits": 1,

  "WriteCapacityUnits": 1

}

Giải thích về các tham số truyền vào:   
+ TableName: Tên của table được khởi tạo

+ HashKeyName: Tên của Partition Key (Bắt buộc)

+ HashKeyType: Kiểu dữ liệu của Partition Key (S|N|B)

+ RangeKeyName: Tên của Sort Key (Không bắt buộc)

# Nếu không truyền vào thì bảng tạo ra sẽ không có Sort Key

+ RangeKeyType: Kiểu dữ liệu của  Sort Key (S|N|B)

+ ReadCapacityUnits: Quy định kích thước đọc tối đa

+ WriteCapacityUnits: Quy định kích thước ghi tối đa

Kết quả trả về: Không có kết quả trả về

**DeleteTable**

https://sqs.us-east-1.amazonaws.com/549480557245/TriggerDeleteTable

Miêu Tả: Xóa một table dựa trên tham số truyền vào (table name)

Ví Dụ Message Body:

{

  "TableName": "Employees"

}

Giải thích các tham số truyền vào:

* TableName: tên table được truyền vào (Employees)

**Create Item**

https://sqs.us-east-1.amazonaws.com/549480557245/TriggerCreateItem

Miêu tả: Thêm một item dựa trên tham số truyền vào

Ví dụ về Message Body:

{

  "TableName": "Students",

  "uid": "012222222",

  "Name": "Trung",

  "Age": 20,

  "Male": true

}

Giải thích tham số truyền vào

TableName: Tên muốn thêm Item

Những cặp key - value còn lại là các thuộc tính của Item và giá trị của chúng

Những cặp key - value là key bắt buộc phải có

Kết quả trả về: Không có kết quả trả về

**Delete Item**

https://sqs.us-east-1.amazonaws.com/549480557245/TriggerDeleteItem

Miêu tả: Xóa item dựa trên các tham số truyền vào

Ví dụ về Message Body:

{

  "TableName": "Students",

  "Key": {

    "uid": "012222222"

  }

}

Giải thích các tham số truyền vào

TableName: tên bảng

Key: Là một danh sách chứa các các cặp key-value là các key của item và giá trị của chúng

#Lưu ý nếu có Range Key thì “Key” sẽ có 2 cặp key-value là 2 key và giá trị của chúng

**Update Item**

https://sqs.us-east-1.amazonaws.com/549480557245/TriggerUpdateItem

Miêu Tả: cập nhật lại các Item trong table dựa trên tham số truyền vào và xác định Item bằng Key

Ví dụ Message Body

{

  "TableName": "Students",

  "Key": {

    "uid": "012222222"

  },

  "Age": 20,

  "Male": true,

  "New": "Trung"

}

Giải thích các tham số truyền vào:

* "TableName": tham số truyền vào là tên table (Students) có Item cần update
* “Key”:  tham số truyền vào danh sách chứa các cặp key-value của Item để xác định dòng Item cần Update
* Những cặp key - value còn lại là các thuộc tính của Item và giá trị của chúng mà cần Update

#Lưu ý nếu có Range Key thì “Key” sẽ có 2 cặp key-value là 2 key và giá trị của chúng.