
Assessment Documentation for Announcement REST API on Cloud Formation, AWS

Prepared and developed by
Saravanakumar R

Page of Index

Requirements and Pre-Requisite

1. Create a free tier account at: https://aws.amazon.com/free/
2. Start Implementing the given assessment in the same account in with given time period
3. Kindly do reach us quickly on any doubts
4. All resources should be created by CloudFormation template only, no manual resource
creation at console.
5. Manually created resource (from console) will not be considered for evaluation
6. After completion of implementation, submit the assessment:
    a. Zip the files:
        i. Source code
        ii. Formation template
        iii. API Contract
        iv. Postman Collection (Export as V2.1)
        v. Simple documentation
    b. Upload the zip fine in any S3 bucket in the same account
    c. Share the AWS Account credentials to us by mentioning S3 bucket name
7. After completion of review you can change your credentials or delete your account.

Description and Layout Architecture

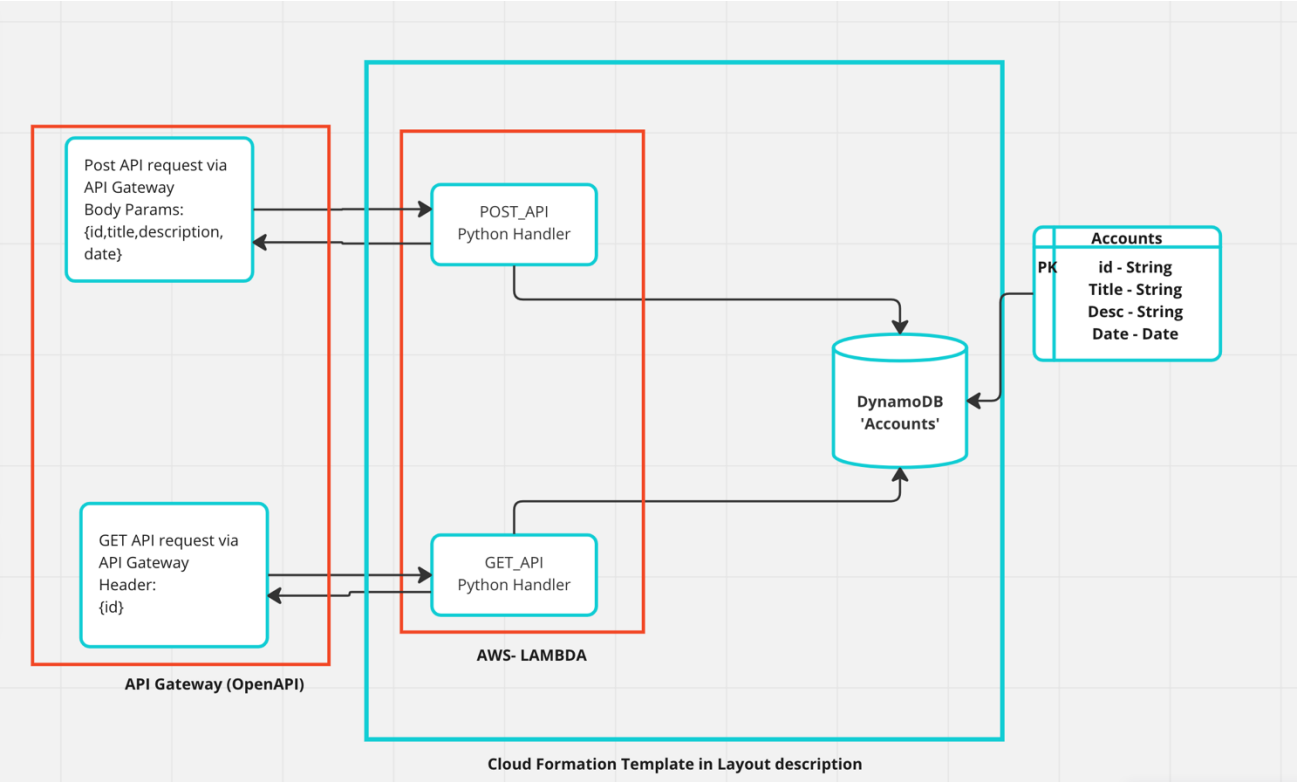

Fig 2.1 The Architecture Layout of Announcement API using AWS Cloud Formation

Objective:

Need to develop API using cloud formation and Open API in AWS, with DynamoDB as the database.

Process:

Here the entire process has been segregated in to two layers which will be created using cloud formation json scripts and open api json scripts, those two scripts has been attached along with this documents in AWS S3 bucket.

The cloud formation json consists the creation of lambda layer for both POST and GET API as python handlers, and along with this auto creation of DynamoDB with id as the primary key, once the json uploaded in cloud stack, the stack will run and create the Lamda , DB automatically and stored as in s3 bucket.

Note : *Please find the attached documents for Cloud formation json and python handlers files in S3 bucket*

The open API json which contains the creation of POST and GET API using APIGateways in AWS, once the API created, need to integrate the corresponding API with already created Lamda functions in Cloud stack.

Below json snippets are the positive test cases used for POST and GET Api's

POST API:

{
'id': 'ABEF-1011-a',
'title': 'Books of JK Rowling',
'description': 'Harry Potter series'
'date': '10-11-2022'
}

GET API:

{
'id': 'ABEF-1011-a'
}

Dynamo DB

The Database has been created using the cloud formation template with ID with datatype String as primary key, along with the API needed title, description and date as the secondary columns as required.

Attached Documents:

1. Cloud Formation Scripts as Json file.
2. OpenAPI scripts as Json file.
3. Architecture Documents.
4. Postman Collections.
5. Source Python code as Python files for reference.