AWStack training - Serverless

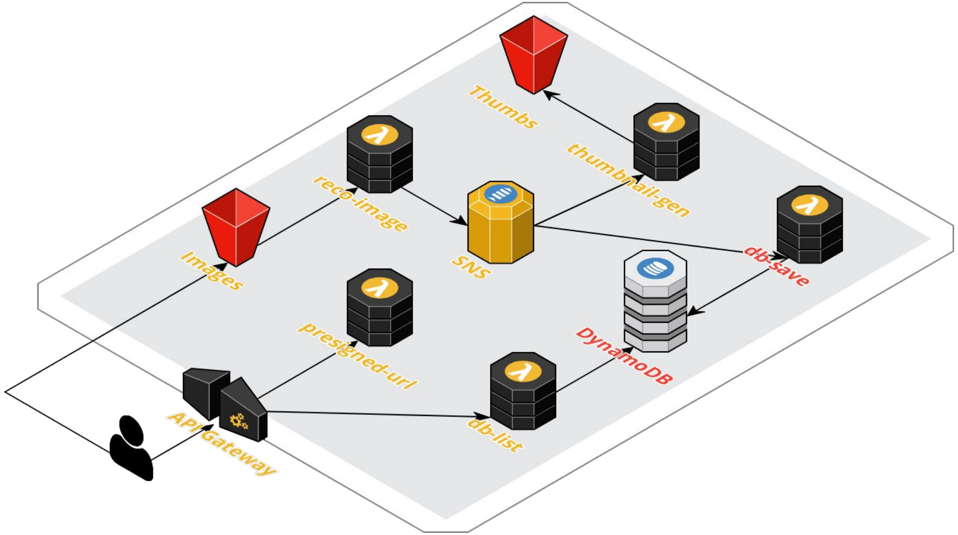
Hands On #5 - DynamoDB - Save results

Overview

This Hands-on is composed of 2 parts:

1. DynamoDB Part : new table to store image recognition results

2. Lambda Part : new function to write results in DynamoDB, triggered by SNS topic



Let’s go! | DynamoDB Part

Go to Virginia region

Create a DynamoDB table having these properties:

- **Table Name**: ImageRecoResults
- **Primary Key**: id
- Use the **default settings**

Once done -> Go to **Lambda Part**

Context:

In this part we create a new DynamoDB table in order to store the image recognition results.

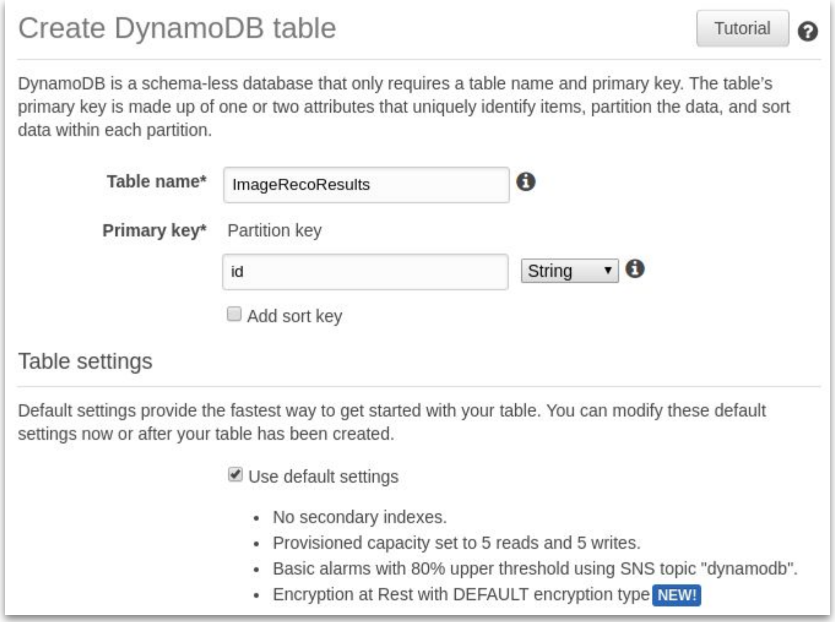
Documentation:

https://docs.aws.amazon.com/dynamodb/index.html

Hint 1

DynamoDB - Create a new table

“ImageRecoResults”



DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* ImageRecoResults

Primary key* Partition key

id String

□ Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☑ Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type NEW!

Let’s go! | Lambda Part

Create a lambda function having these properties:

- **Name**: py-aws-lambda-db-save
- **Runtime**: Python 3.6
- **Trigger**: SNS (use the topic created previously)
- **Role**: serverless_lambda_role
- Add a new **Environment variable** containing the table name created previously
  - Name: dynamodbTableName
  - Value: ImageRecoResults
- Upload the **Function code** from the S3 bucket: https://s3.amazonaws.com/awstacktraining-serverless-resources/code-templates/py-aws-lambda-db-save-template.zip

Once done -> Go to **Testing** part to test your Lambda

Context:

In this part we create a new Lambda function in charge of writing recognition results in DynamoDB.

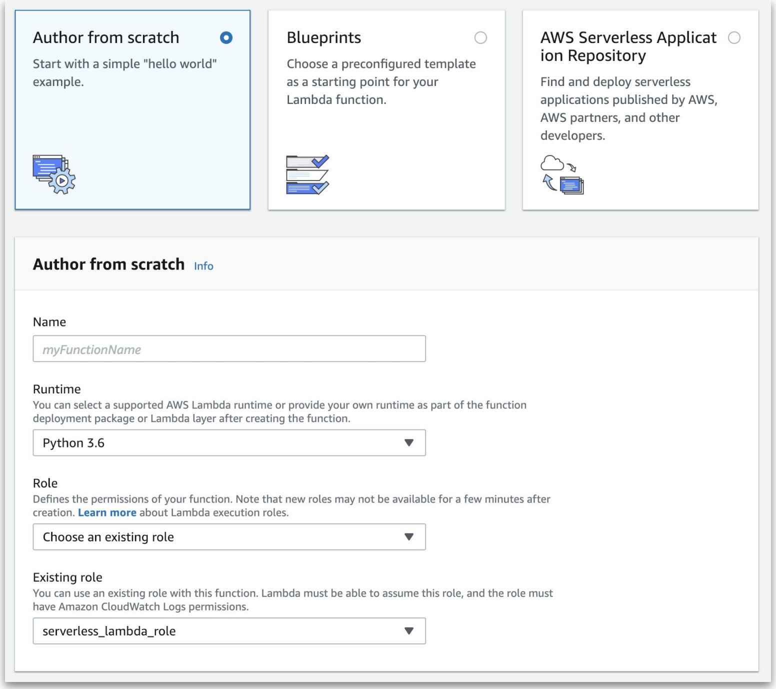
The function is triggered by each publish in the SNS results topic created previously.

Documentation:

https://boto3.amazonaws.com/v1/documentation/api/latest/guide/dynamodb.html

Hint 2

Lambda Creation - create a new Lambda function "py-aws-lambda-db-save" using the existing role "serverless_lambda_role"



● Author from scratch

Start with a simple "hello world" example.

○ Blueprints

Choose a preconfigured template as a starting point for your Lambda function.

○ AWS Serverless Applicat ion Repository

Find and deploy serverless applications published by AWS, AWS partners, and other developers.

Author from scratch Info

Name

myFunctionName

Runtime

You can select a supported AWS Lambda runtime or provide your own runtime as part of the function deployment package or Lambda layer after creating the function.

Python 3.6 ▼

Role

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. **Learn more** about Lambda execution roles.

Choose an existing role ▼

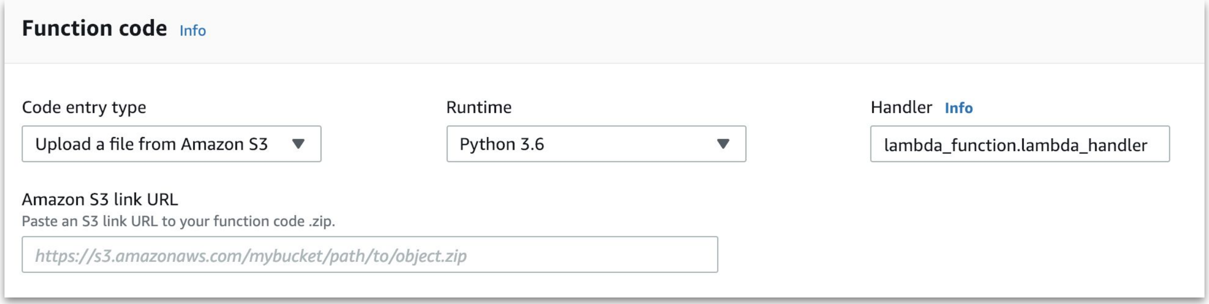
Existing role

You can use an existing role with this function. Lambda must be able to assume this role, and the role must have Amazon CloudWatch Logs permissions.

serverless_lambda_role ▼

Hint 3

Lambda Configuration - Upload the function code from the S3 link URL given in “Let’s Go” section



Function code Info

Code entry type: Upload a file from Amazon S3 ▼

Runtime: Python 3.6 ▼

Handler Info: lambda_function.lambda_handler

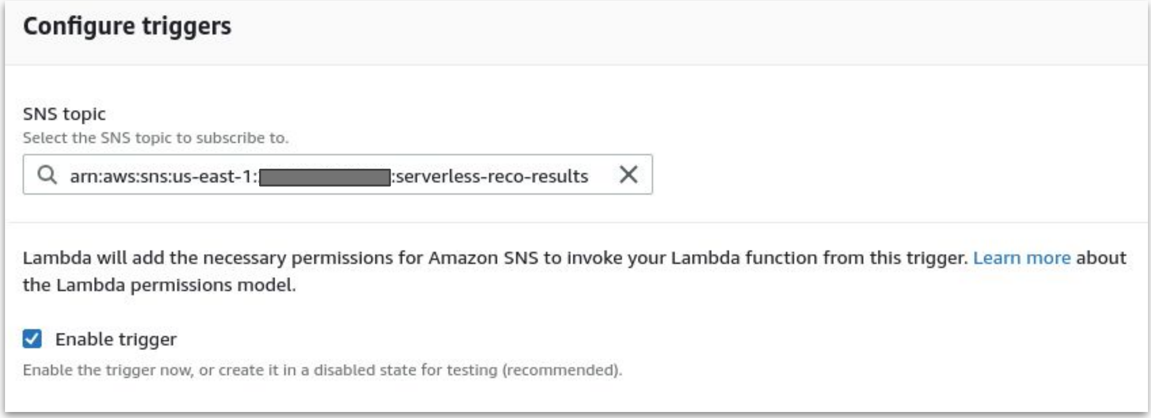
Amazon S3 link URL

Paste an S3 link URL to your function code .zip.

https://s3.amazonaws.com/mybucket/path/to/object.zip

Hint 4

Lambda Configuration - Add a new SNS Trigger from list on the left



Configure triggers

SNS topic

Select the SNS topic to subscribe to.

arn:aws:sns:us-east-1:[redacted]:serverless-reco-results

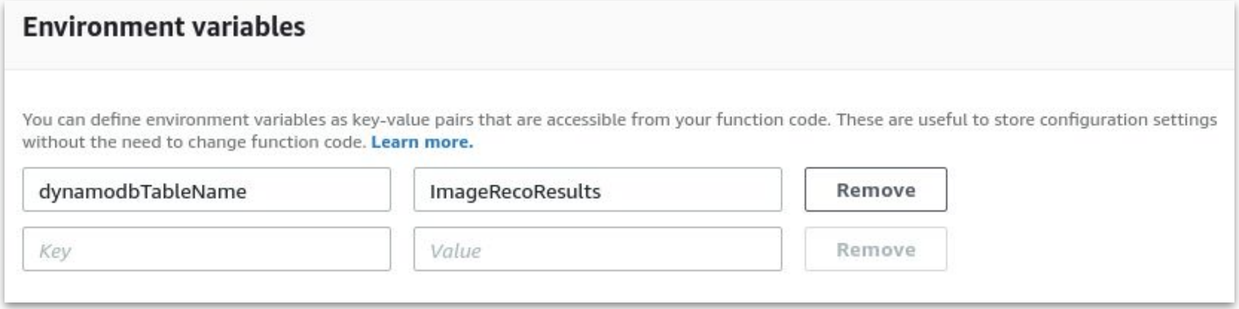
Lambda will add the necessary permissions for Amazon SNS to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.

☑ Enable trigger

Enable the trigger now, or create it in a disabled state for testing (recommended).

Hint 5

Lambda Configuration - Add a new environment variable “dynamodbTableName”



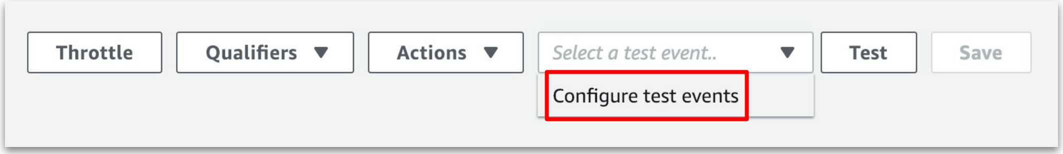
Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more.

dynamodbTableName	ImageRecoResults	Remove
Key	Value	Remove

Test the lambda function

1. Copy the test json sample **test-sample.json** from **Function code**
2. Configure a new **Test event** from the top menu



3. Paste the json sample, create the test event and test !

Done !

Test the full integration by uploading an image in the S3 bucket, checking CloudWatch logs and DynamoDB table !

You can download the Lambda code at

https://github.com/laurentnoireterre/py-aws-lambda-db-save