

AWS training - Serverless

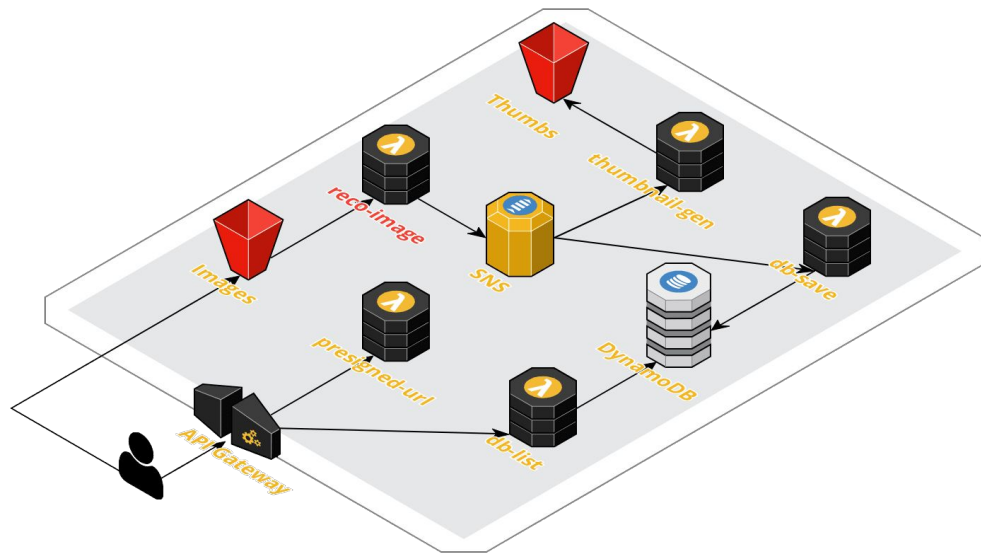
Hands On #3 - Lambda - Image recognition



Overview

This Hands-on has only one part:

1. [Lambda Part](#) : new function to analyze and recognize images



Let's go!

Go to Virginia region

N. Virginia ▼

Create a lambda function having these properties:

- **Name:** py-aws-lambda-image-reco
- **Runtime:** Python 3.6
- **Trigger:** S3 (**all create events** on the bucket created previously)
- **Role:** serverless_lambda_role
- **Layer:** *Tensorflow-Kera-Pillow* layer **using this ARN**
`arn:aws:lambda:us-east-1:347034527139:layer:tf_keras_pillow:3`
- **Memory & Timeout:** adjust the memory to **512MB** and timeout to **10 sec.**
- Upload the **Function code** from the S3 bucket:
<https://s3.amazonaws.com/awstacktraining-serverless-resources/code-templates/py-aws-lambda-image-reco-template.zip>

Once done -> Go to [Testing part](#) to test your Lambda

Context:

In this part we create a lambda function in charge of analyzing an image. This function is triggered by S3 for each image uploaded in the bucket we created previously.

We integrate Keras deep learning API to our function using a shared Lambda layer named *Tensorflow-Kera-Pillow*

Documentation:

<https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html>


<https://github.com/antonpaquin/Tensorflow-Lambda-Layer>

Hint 1

Lambda Creation - create a new
Lambda function
“py-aws-lambda-image-reco” 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 Application Repository ☐

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



Author from scratch [Info](#)

Name

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.

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.

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.

Hint 2

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 3

Lambda Configuration - Add a new
S3 Trigger from list on the left

Configure triggers

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

serverless-training-img ▼

Event type
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events ▼

Prefix
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

e.g. images/

Suffix
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

e.g. .jpg

Lambda will add the necessary permissions for Amazon S3 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 4

Lambda Configuration - Add the layer

“arn:aws:lambda:us-east-1:347034527139:layer:tf_keras_pillow:3” to the function

Layer selection

Select an existing AWS-vended layer or layer in your account, or provide a layer that has been shared with you. You can connect a maximum of 5 layers to a function.

- ☐ Select from list of runtime compatible layers
- ☒ Provide a layer version ARN

Provide a layer version ARN

Layer version ARN [Info](#)

Provide the ARN of a layer to add to your function.

arn:aws:lambda:us-east-1:347034527139:layer:tf_keras_pillow:3

Hint 5

Lambda Configuration - Modify the lambda basic settings

Basic settings

Description

Memory (MB) [Info](#)

Your function is allocated CPU proportional to the memory configured.

512 MB

Timeout [Info](#)

0

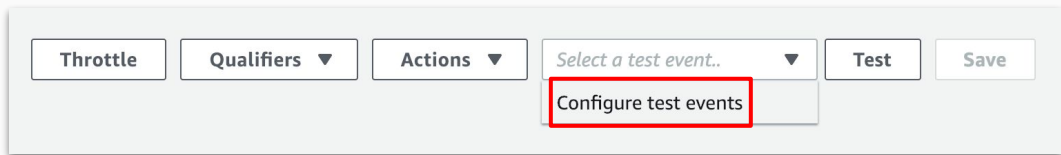
min

10

sec

Test the lambda function

1. Upload an image in the S3 bucket from the console
2. Copy the test json sample **test-sample.json** from **Function code**
3. Configure a new **Test event** from the top menu



4. Paste the json sample and replace `<YOUR_BUCKET_NAME>` and `<YOUR_BUCKET_TEST_IMAGE>` by your bucket and image names
5. Create the test event and test !

Done !

Test the full integration with S3 by uploading an image and checking
CloudWatch logs

You can download the answer code at



<https://github.com/laurentnoireterre/py-aws-lambda-image-reco>