AWStack training - Serverless

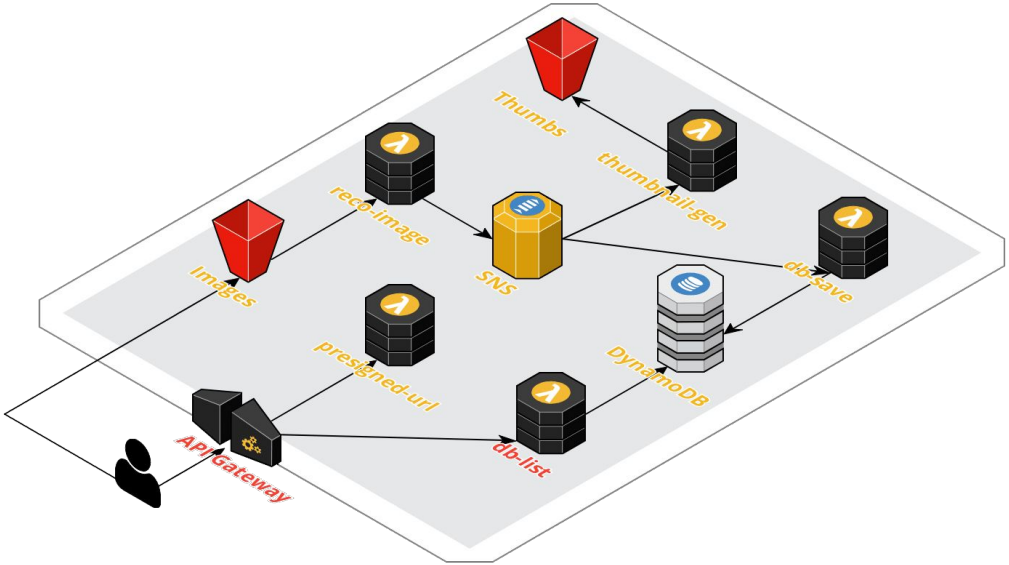
Hands On #6 - API Gateway | Lambda - List items

Overview

This Hands-on is composed of 2 parts:

1. Lambda Part : new function to list the recognition result items stored in DynamoDB

2. API Gateway Part : new GET method on existing API Gateway to route the ingress traffic to the Lambda function



Let’s go! | Lambda Part

Go to Virginia region N. Virginia

Create a lambda function having these properties:

- **Name**: py-aws-lambda-db-list-items
- **Runtime**: Python 3.6
- **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-list-items-template.zip

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

Context:

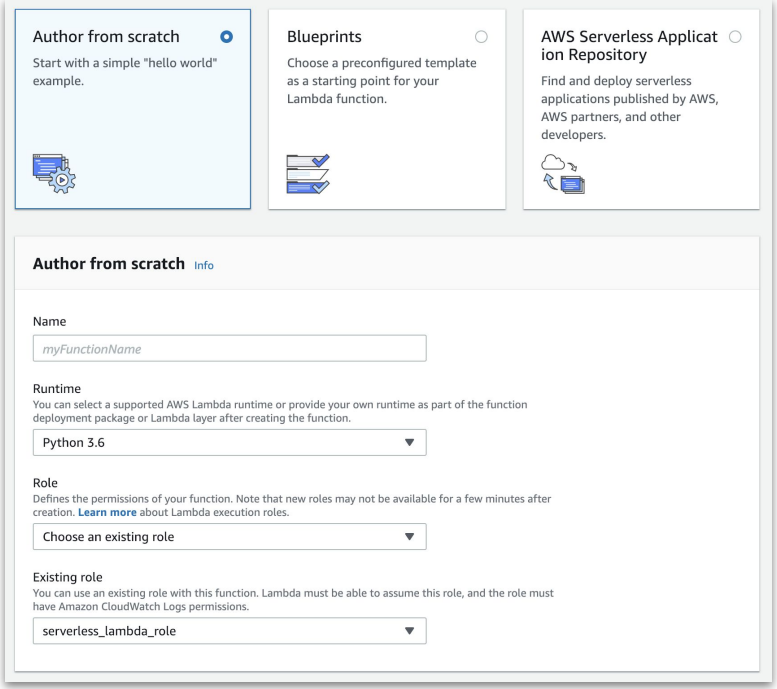
In this part we create a new Lambda function in charge of listing the recognition result items stored in DynamoDB.

Documentation:

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3.html

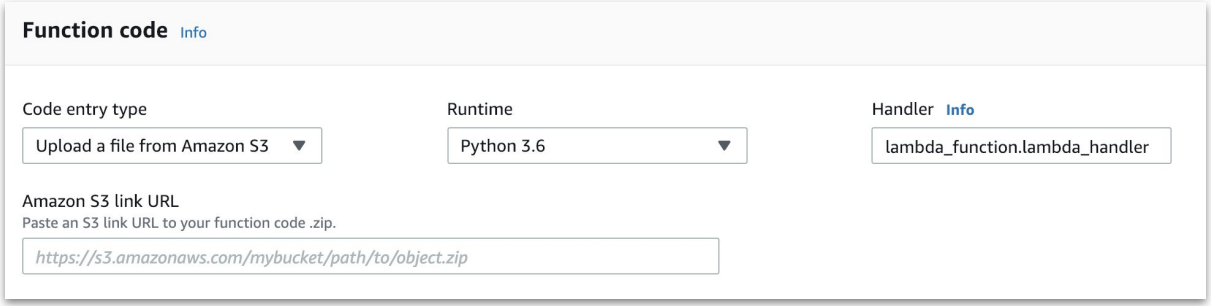
Hint 1

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



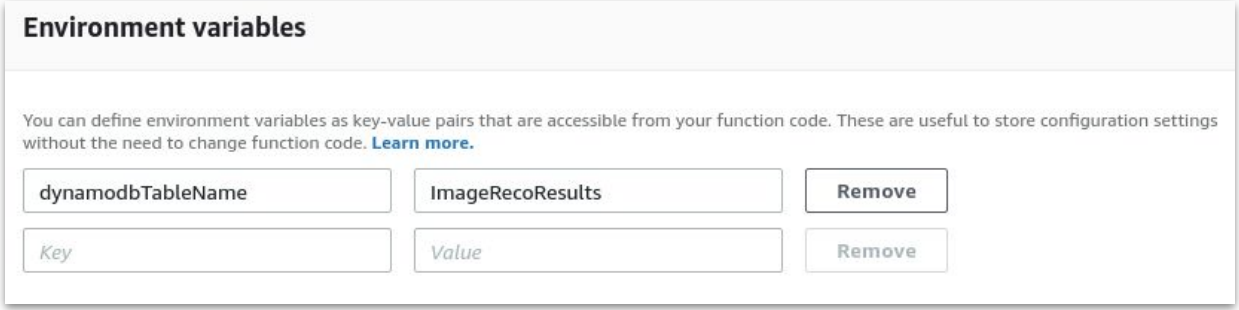
Hint 2

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



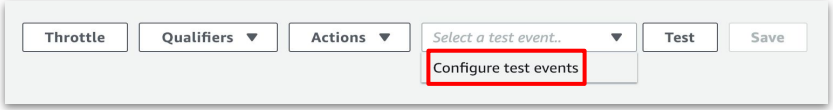
Hint 3

Lambda Configuration - Add a new environment variable "dynamodbTableName"

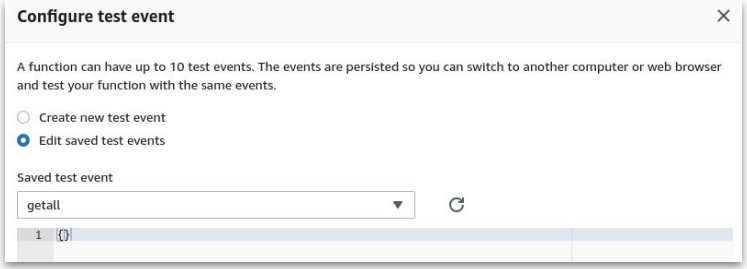


Test the lambda function

1. Configure a new **Test event** from the top menu



2. Set an empty JSON content, create the test event and test !



Let’s go! | API Gateway Part

Create a new API resource on “image-reco” API Gateway, having these properties:

- **Name**: results
- **Resource**: /results
- Create a new **GET** method
- **Lambda function**:
- **CORS** has to be enabled
- Use **Lambda proxy integration**
- Integrate with the **py-aws-lambda-db-list** lambda function

Before deploying -> Go to **Testing** part to test the API endpoint

- Deploy the API on the **dev** stage

After deploying -> Go to **Done !** part to test the full integration

Context:

In this part we create the API Gateway resource in front of the lambda. The proxy mode will route the ingress traffic directly to the function without any modification.

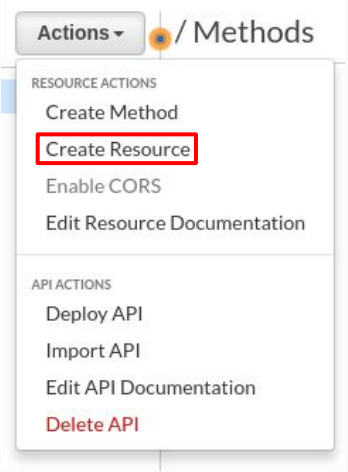
Documentation:

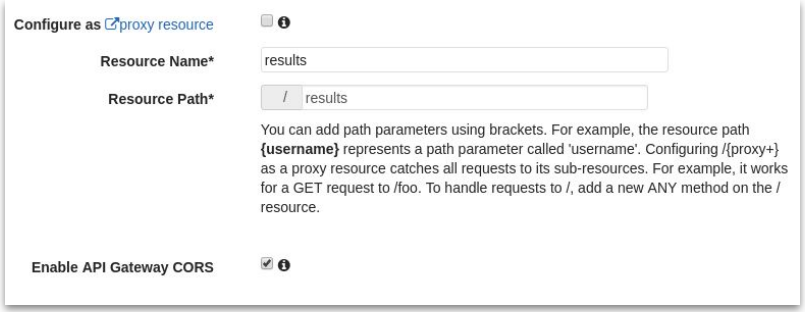
https://docs.aws.amazon.com/fr_fr/lambda/latest/dg/with-s3-example-deployment-pkg.html#with-s3-example-deployment-pkg-python

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3.html

Hint 4

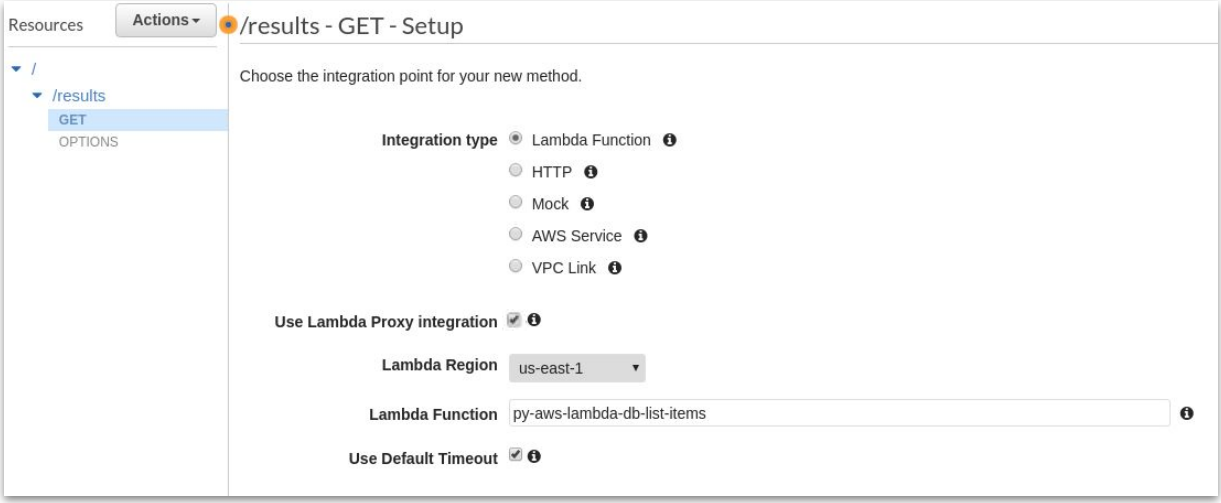
API Gateway - Open the API Gateway “reco-image” and create a new resource “results”, with CORS enabled





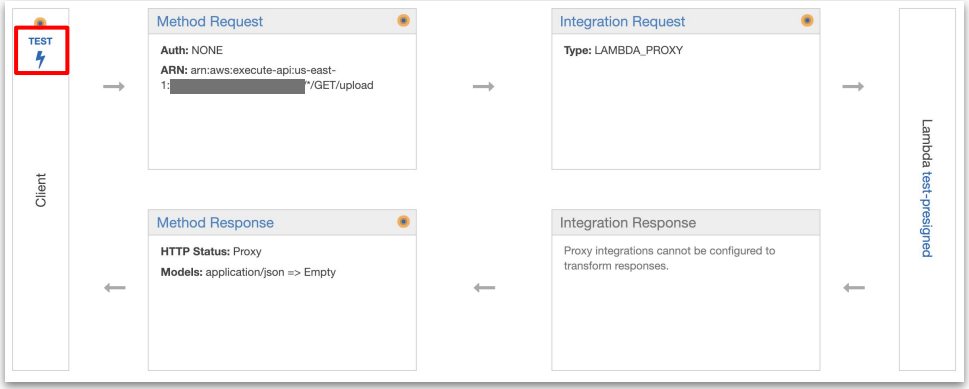
Hint 5

API Gateway - create a new GET method pointing to the lambda function created previously. Use the proxy mode integration



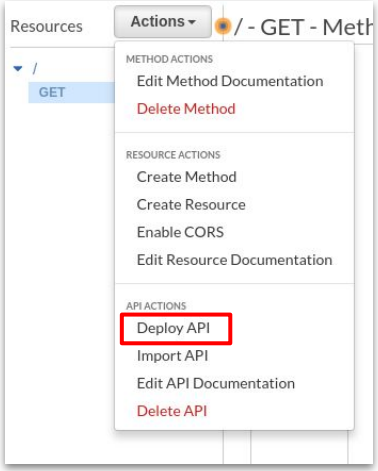
Test the API Gateway resource

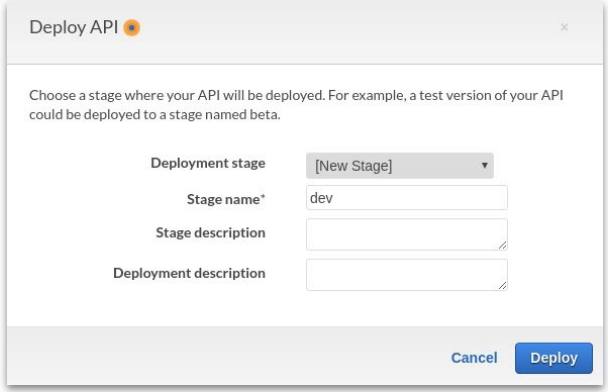
API Gateway - Test the GET method execution



Hint 6

API Gateway - Deploy the API to the stage “dev”





Done !

Test the API endpoint using the invoke URL !

<YOUR_API_INVOKE_URL>/results

You can download the Lambda code at

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