AWS

Using Lambda Function

Microservices Architecture (Introduction)

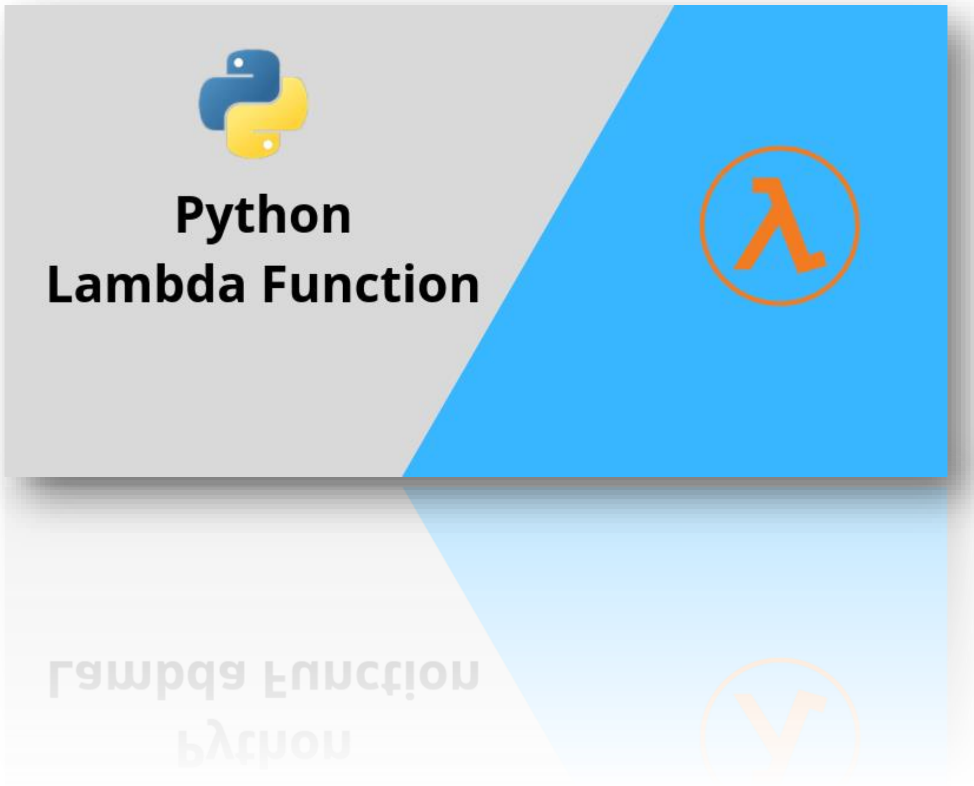
Teacher: Mr. Zheng Li

Student: Juan Albornoz

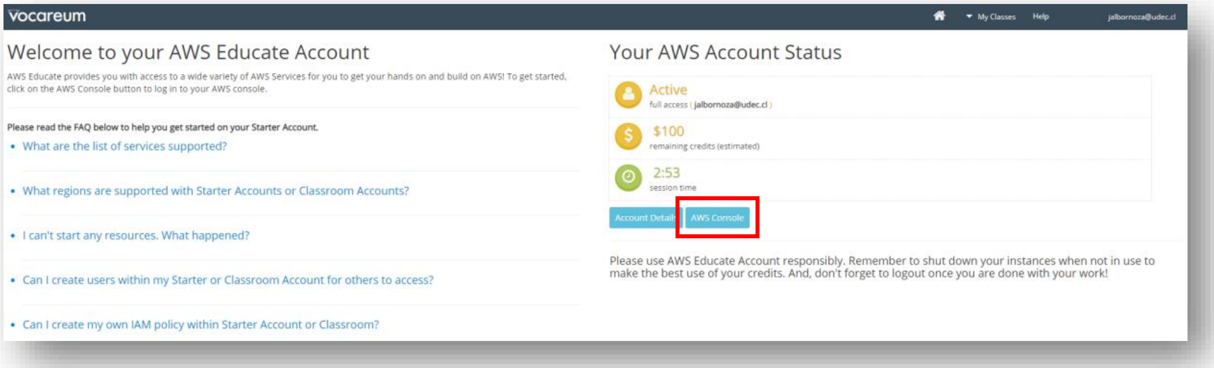
¿What’s the lambda function and, how it’s useful?

The cloud usage model it’s well provided to handle tasks in a lot of scenarios by the manage and the use of the resources along with many other servers but, ¿what if you’d had a way to bind services together for attending certain tasks by triggering the execution of a single code?. Here is where Python Lambda Functions could be very important.

Lambda provides runtimes for Python by executing your code to process events. The environment includes SDK for Python, more specifically Python 3.7.



1. The first step after logging into the AWS interface it´s to open the *AWS Console* showed under your account status information.



Welcome to your AWS Educate Account

AWS Educate provides you with access to a wide variety of AWS Services for you to get your hands on and build on AWS! To get started, click on the AWS Console button to log in to your AWS console.

Please read the FAQ below to help you get started on your Starter Account.

- What are the list of services supported?
- What regions are supported with Starter Accounts or Classroom Accounts?
- I can't start any resources. What happened?
- Can I create users within my Starter or Classroom Account for others to access?
- Can I create my own IAM policy within Starter Account or Classroom?

Your AWS Account Status

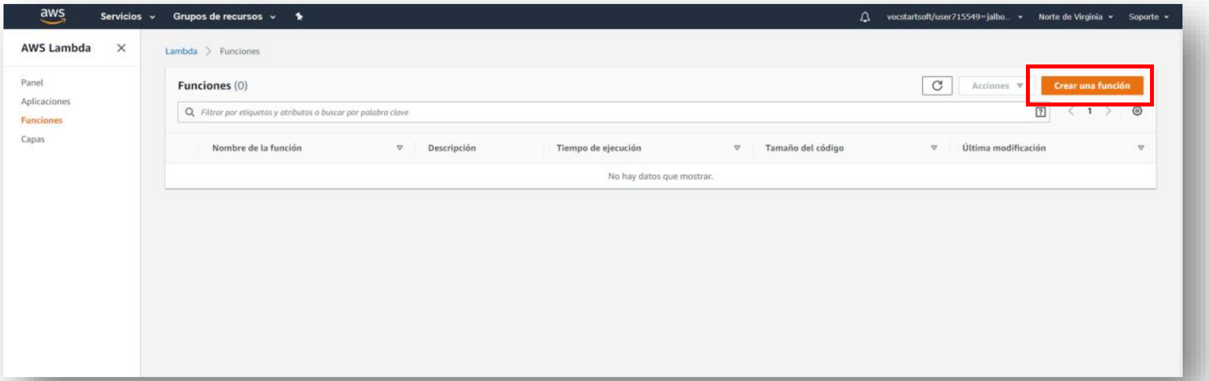
Active
full access (jalbornoza@udec.cl)

$100
remaining credits (estimated)

2:53
session time

Please use AWS Educate Account responsibly. Remember to shut down your instances when not in use to make the best use of your credits. And, don't forget to logout once you are done with your work!

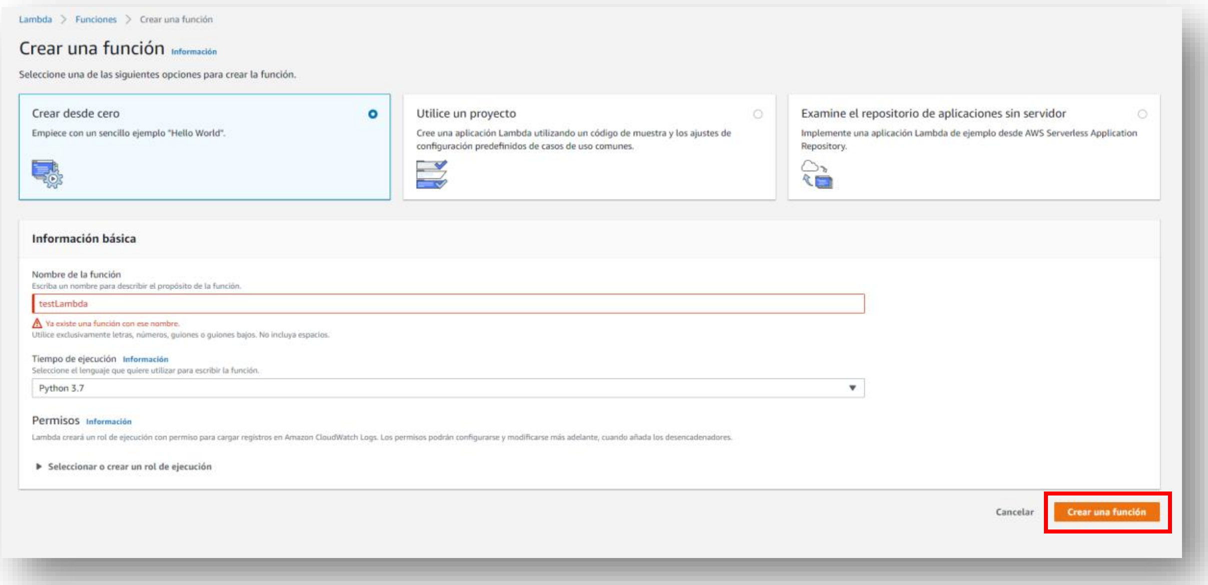
2. After that you should see the interface for Lambda. You have an empty list of functions, so let's create a test one.



Funciones (0)

Nombre de la función	Descripción	Tiempo de ejecución	Tamaño del código	Última modificación
No hay datos que mostrar.				

3. Put a name to your function and the right language running it (Python 3.7 in our case). My function it's named *testLambda.*



Lambda > Funciones > Crear una función

Crear una función Información

Seleccione una de las siguientes opciones para crear la función.

● Crear desde cero
Empiece con un sencillo ejemplo "Hello World".

○ Utilice un proyecto
Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predefinidos de casos de uso comunes.

○ Examine el repositorio de aplicaciones sin servidor
Implemente una aplicación Lambda de ejemplo desde AWS Serverless Application Repository.

Información básica

Nombre de la función
Escriba un nombre para describir el propósito de la función.
testLambda
⚠ Ya existe una función con ese nombre.
Utilice exclusivamente letras, números, guiones o guiones bajos. No incluya espacios.

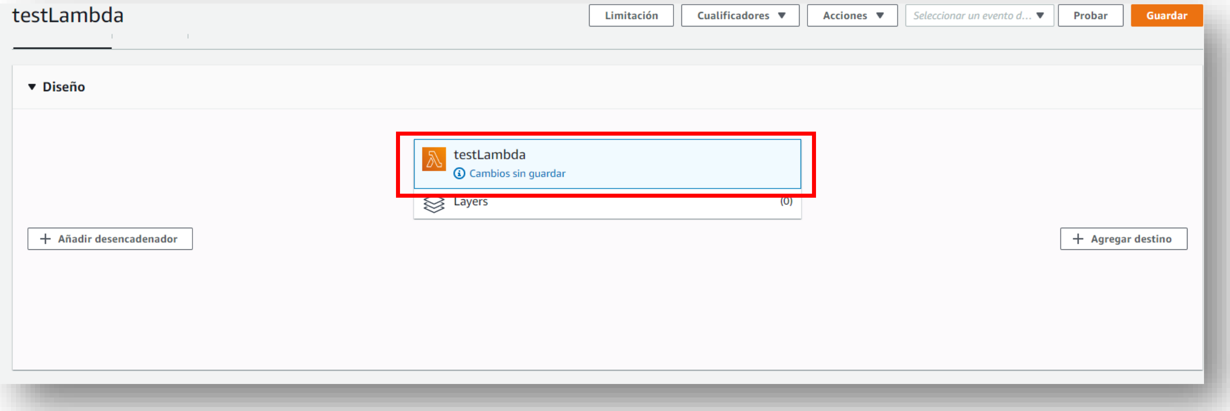
Tiempo de ejecución Información
Seleccione el lenguaje que quiere utilizar para escribir la función.
Python 3.7

Permisos Información
Lambda creará un rol de ejecución con permiso para cargar registros en Amazon CloudWatch Logs. Los permisos podrán configurarse y modificarse más adelante, cuando añada los desencadenadores.

▶ Seleccionar o crear un rol de ejecución

Cancelar | Crear una función

4. After creating your function we have all the configurations for our function. In the *Design* part we have our function alone for the moment. Maybe later we´re going to use *triggers* and use different *layers* for our functions. Also there is a text editor for coding in the same interface without uploading any code from your computer.



testLambda

Limitación | Cualificadores ▼ | Acciones ▼ | Seleccionar un evento d... ▼ | Probar | Guardar

▼ Diseño

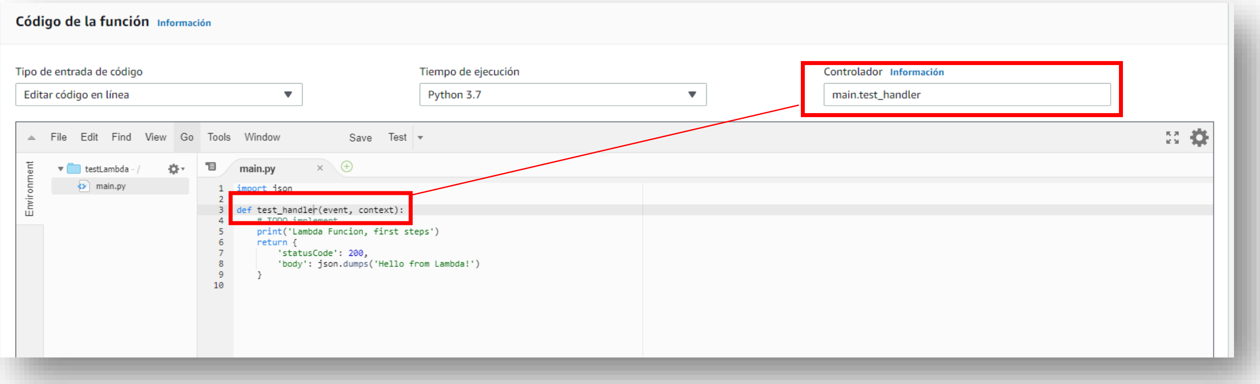
testLambda
ⓘ Cambios sin guardar

Layers (0)

+ Añadir desencadenador

+ Agregar destino

5. Here it´s important to notice that we have to rename our *Controller* to the name of our defined function (or method) written in the *.py* for a successful execution. In my case I named it *test_handler()* so, the *Controller* it's re-named as the same.



Tipo de entrada de código: Editar código en línea

Tiempo de ejecución: Python 3.7

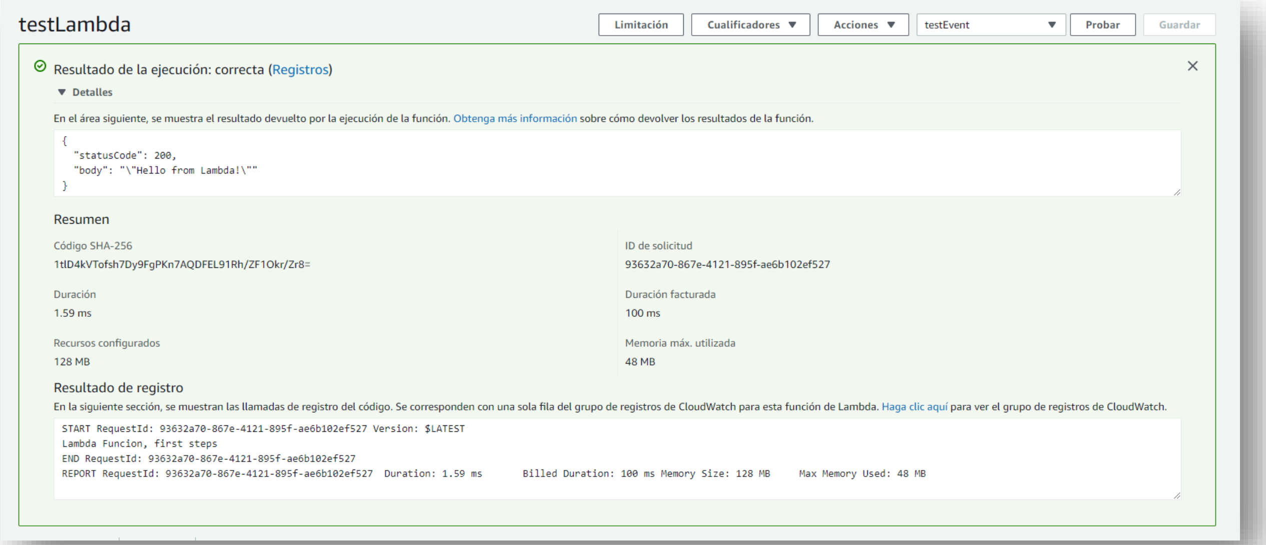
Controlador: main.test_handler

```
import json

def test_handler(event, context):
    # TODO implement
    print('Lambda Funcion, first steps')
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }

```

6. Finally we have to execute our function to test it so, in the upper side of our interface there are the buttons for *Save* changes and *Test* for running.



Resultado de la ejecución: correcta (Registros)

En el área siguiente, se muestra el resultado devuelto por la ejecución de la función. Obtenga más información sobre cómo devolver los resultados de la función.

```
{
  "statusCode": 200,
  "body": "\"Hello from Lambda!\""
}
```

Resumen

Código SHA-256: 1tlD4kVTofsh7Dy9FgPKn7AQDFEL91Rh/ZF1Okr/Zr8=

ID de solicitud: 93632a70-867e-4121-895f-ae6b102ef527

Duración: 1.59 ms

Duración facturada: 100 ms

Recursos configurados: 128 MB

Memoria máx. utilizada: 48 MB

Resultado de registro

En la siguiente sección, se muestran las llamadas de registro del código. Se corresponden con una sola fila del grupo de registros de CloudWatch para esta función de Lambda. Haga clic aquí para ver el grupo de registros de CloudWatch.

```
START RequestId: 93632a70-867e-4121-895f-ae6b102ef527 Version: $LATEST
Lambda Funcion, first steps
END RequestId: 93632a70-867e-4121-895f-ae6b102ef527
REPORT RequestId: 93632a70-867e-4121-895f-ae6b102ef527  Duration: 1.59 ms       Billed Duration: 100 ms Memory Size: 128 MB     Max Memory Used: 48 MB
```