

1- Crear RDS

1-1 Entrando desde el panel web a la sección RDS desde el buscador y luego click en **Create database** -> seleccionar tipo Standar -> MySQL -> Marcar si o si "Free Tier" -> en Settings en **DB identifier** escribir un nombre, por ej: "MySQLForLambdaMundose" -> en credentials settings usar **Master username y master password**

User: admin

pass: qwerty.1234

Dejar seleccionado la opción "Don't connect to an EC2..."

Seleccionar una VPC sí o sí con más de una zona de disponibilidad y con más de una subnet (puede ser la default) -> public access **No**

Additional configuration -> database options -> initial database name "ExampleDB"

Muestra un estimado de 14USD pero recordar que es mensual, a fines del laboratorio no debería generar costos elevados.

1-2 Una vez creada la RDS la seleccionamos y vamos al desplegable de **actions** y seleccionamos **"set up Lambda connection"** -> crear nueva función -> name "LambdaFunctionRDSmundose" -> bajar y seleccionar en **rds proxy** -> crear nuevo proxy -> seleccionar en database credentials **"Database user and pass"** y colocar debajo las credenciales recién creadas -> por último **set up**.

2- Crear rol/es para Lambdas.

2-1 Ir a IAM por el buscador, después en el panel de la izquierda buscar **Roles**

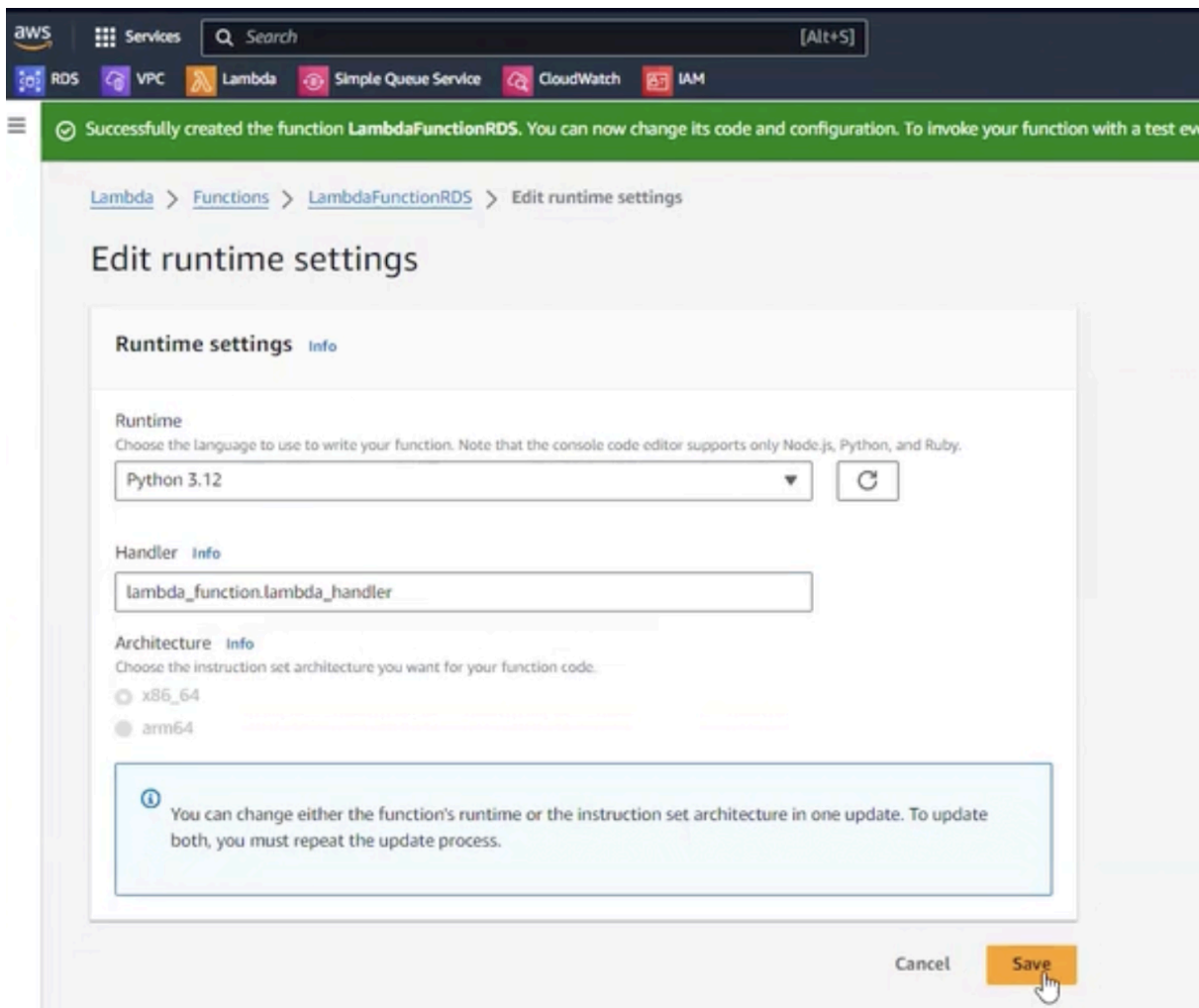
A la derecha de la pantalla click en **Create role** -> seleccionar en **Trusted entity type** "AWS Service" -> en **Use Case** en el desplegable select "Lambda" -> **Next** -> en **Permissions policies** buscar "AWSLambdaSQSQueueExecutionRole" y "AWSLambdaVPCLAccessExecutionRole" -> **Next** -> dar nombre al rol "lambda-vpc-sqs-role-mundose" -> **create role**

Copiar el ARN del rol dsp que se crea.

3- Crear lambda

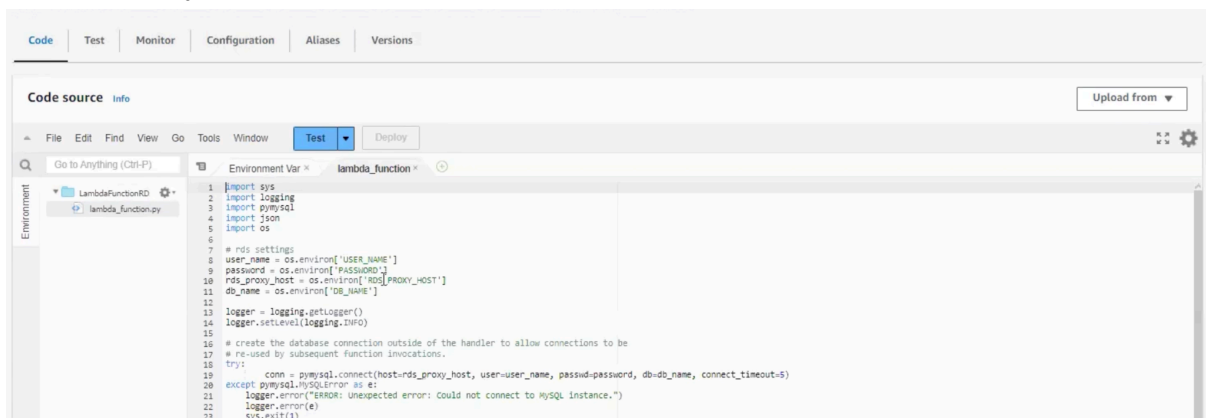
Técnicamente ya debería estar creada, si no aparece dar a refresh de la consola. Si el paso 1-2 esta bien una Lambda fue creada con el nombre que le dimos. Si no aparece dar click en el botón refresh.

Una vez que aparece la seleccionamos y bajamos hasta **"Runtime Settings"** y seleccionamos **"Editar"** (el que es solo editar) y del desplegable **"Runtime"** seleccionamos cualquiera de "Python 3.12" y en **"Handler"** escribimos "lambda_function.lambda_handler" -> **save**

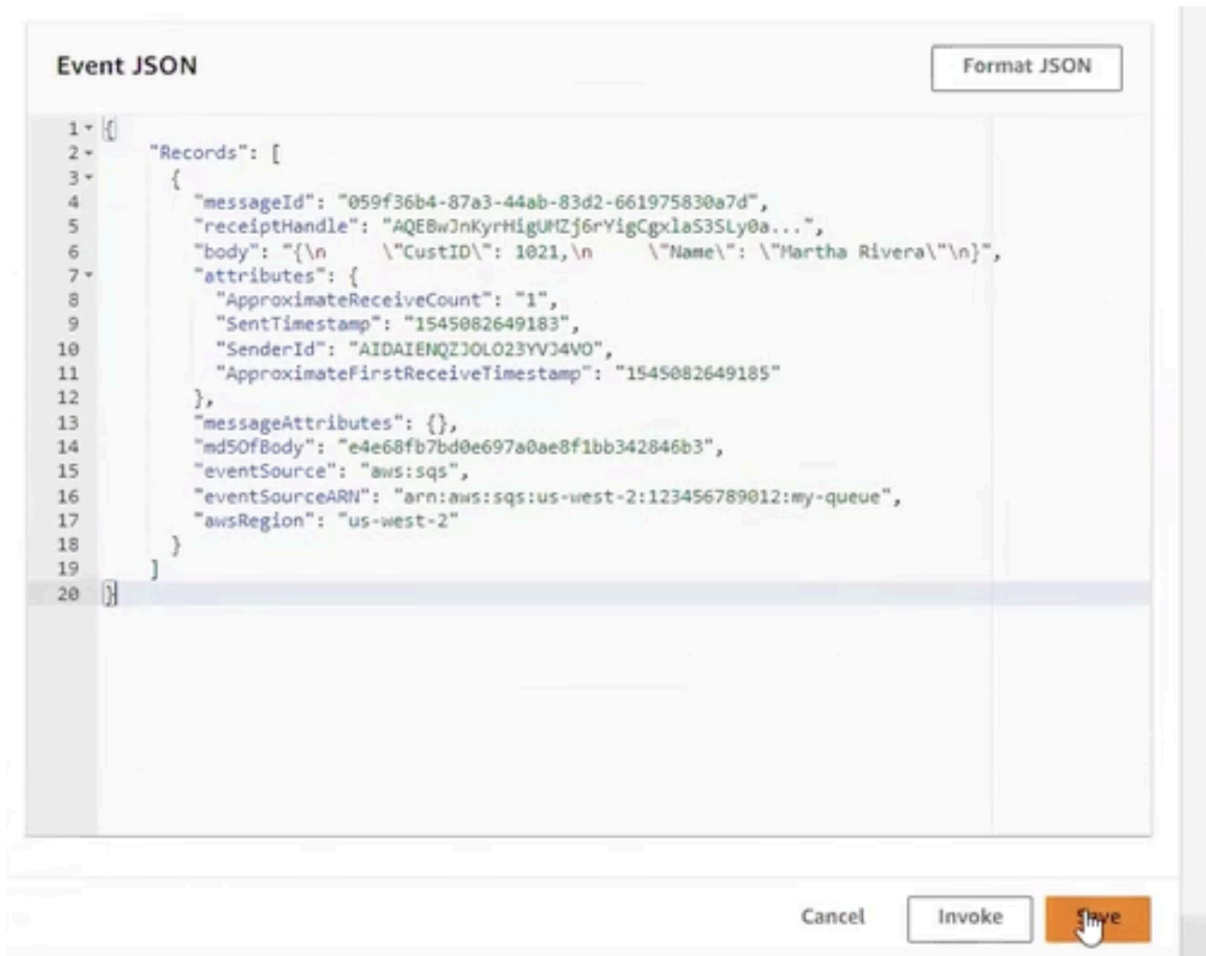


Ahora debemos de subir un .zip con el nombre “lambda_function” (respetando el nombre), este archivo está alojado en el github compartido con el nombre lambda_funtion.py lo transformamos en zip localmente y volvemos al panel web de AWS.

En la pestaña “**Code**” a la misma línea que “**Code source**” en el margen derecho el botón “upload from” y seleccionamos lambda_function.zip (nos debería quedar así)



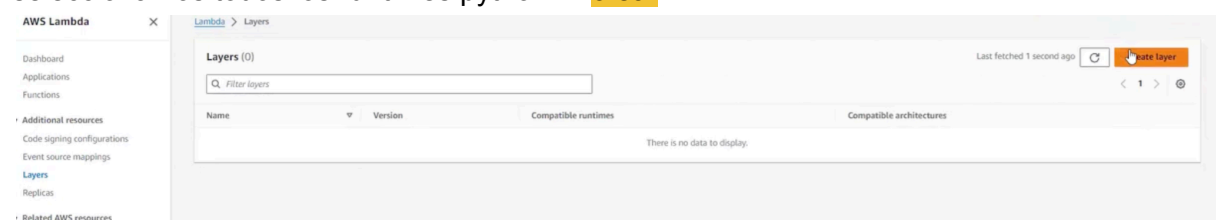
vamos a probar el test (click en **test**) y configuramos event name random por ej. "TestLambada", y pegamos el primer bloque de código del archivo queries.json que está en el repo de github en "Event JSON" -> **Save** y despues Test (va a fallar)



En general funciona de una, pero acá falla por un tema python y lambda. Se soluciona agregando un paso extra, que seria basicamente una layer en nuestra lambda, esto se hace así:

Antes de crear la layer se debe de respetar esta estructura de carpetas que está en el .zip del repositorio básicamente se debe de tener una carpeta "python" y dentro de ella lo necesario para las layer, o sea las dependencias. Sino se quiere usar el zip del repositorio lo hacemos de la siguiente manera "mkdir python -> pip install - -target python pymysql"; para luego hacer esa carpeta python en un .zip

Volviendo a la web de AWS en el panel de la izquierda del servicio de Lambda buscar la opción "**layers**" -> **create layer** -> name "python" -> upload zip recién creado -> y seleccionamos todos los runtimes python -> **crear**



así nos debería de quedar todo

Layer configuration

Name

Description - optional

☒ Upload a .zip file
☐ Upload a file from Amazon S3

python.zip
127.64 KB

For files larger than 10 MB, consider uploading using Amazon S3.

Compatible architectures - optional [Info](#)
Choose the compatible instruction set architectures for your layer.

☐ x86_64
☐ arm64

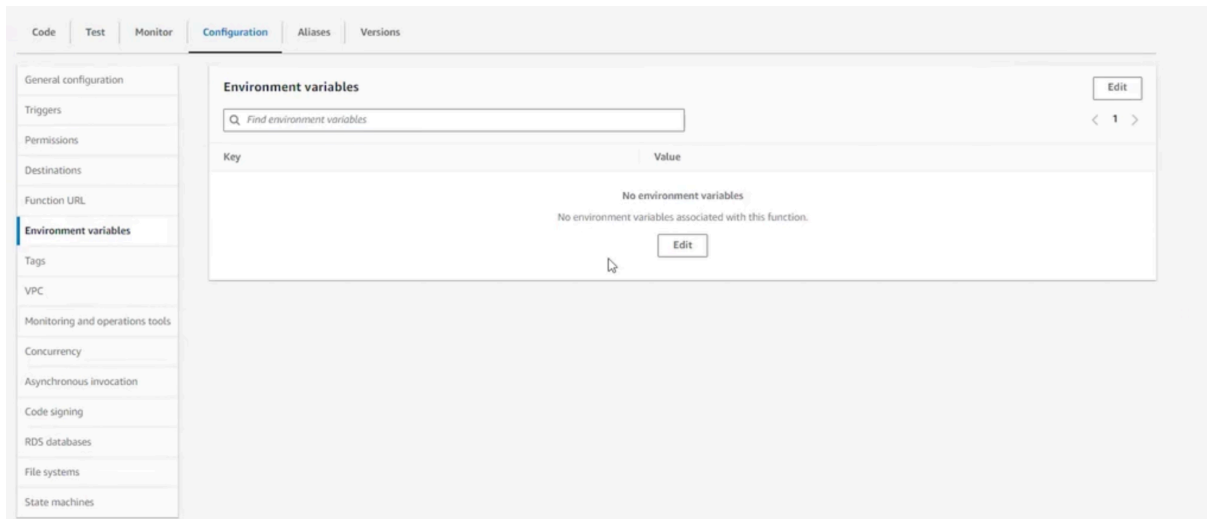
Compatible runtimes - optional [Info](#)
Choose up to 15 runtimes.

License - optional [Info](#)

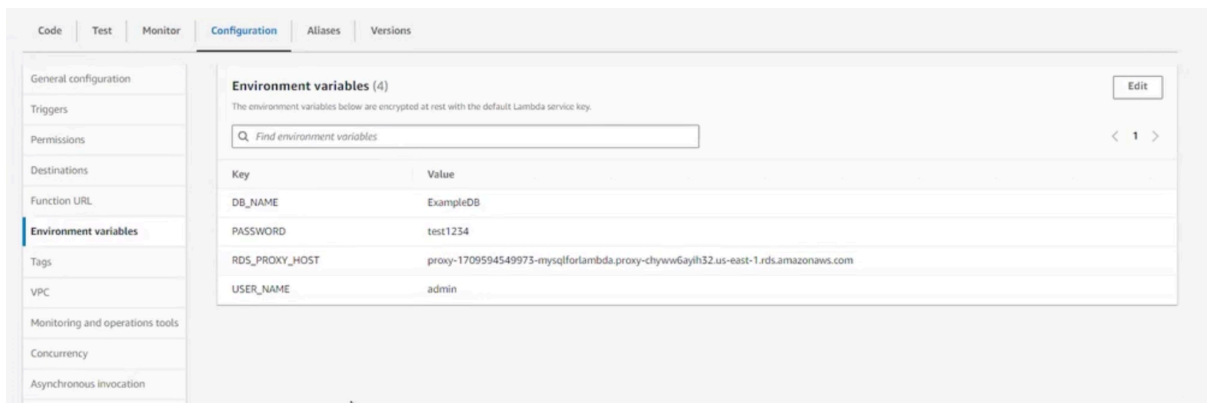
Volvemos a Lambda en el panel de la izquierda y vamos a agregar la layer, scroll hasta el final de la pagina que esta el bloque de **Layer** -> seleccionar "add layer" -> de tipo "custom layer" en las opciones -> y en el desplegable seleccionar la layer creada y la versión correspondiente (se verifica en la sección Layers).

Vamos a la tab "**Configuration**" -> y a la izquierda esta la opcion de variables de entorno -> editar -> add -> vamos agregar las key value:

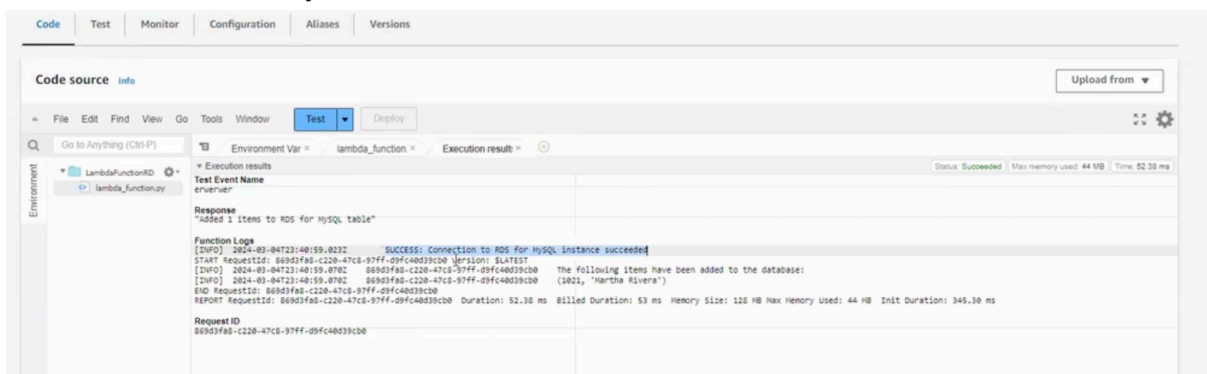
USER_NAME : admin
PASSWORD : qwerty.1234
RDS_PROXY_HOST : (copiar el del proxy creado)
DB_NAME : ExampleDB



Asi debería verse despues de cargar las variables



Vamos a darle a TEST y debería correr.



En la imagen se ve el "SUCCESS" y dos líneas debajo los datos agregados a la base.