



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Entrega 5

Taller - 2022-1

Introducción a Cloud Computing con AWS

28 de abril de 2022 - v1.0

Índice

1. Crear función lambda	5
2. Configuración	6
3. Implementación	8
4. Ejecución	9
5. X-Ray	10

Índice de tablas

1. Mediciones en la ejecución de las diferentes pruebas realizadas	10
--	----

Índice de figuras

1. Captura de pantalla de la “AWS Management Console” posicionada en el servicio Lambda vista de Funciones	5
2. Captura de pantalla de “Create function” antes de crear la función	5
3. Captura de pantalla mostrando la configuración de la función a 512[MB] de memoria y un timeout de 1[s]	6
4. Captura de pantalla mostrando la habilitación de tracing con X-Ray	6
5. Captura de pantalla mostrando el resultado de la ejecución del test por omisión en cold-start (Duración facturada 2[ms], duración Cold-Start 1.34[ms] y Memoria usada 54[MB]	7
6. Captura de pantalla mostrando el resultado de la ejecución del test por omisión en warm-start (Duración facturada 2[ms] y Memoria usada 55[MB]	7

7.	Captura de pantalla mostrando la re-configuración de la función a 128[MB] de memoria y un timeout de 1[s] optimo para el testeo de la ejecución ejemplo	7
8.	Capturas de pantalla mostrando las 3 pruebas consideradas	8
9.	Captura de pantalla mostrando el código desplegado y listo para ejecutar en la consola de AWS	9
10.	Trazas de Cold-Start y Warm-Start del Test 1	10
11.	Trazas de Cold-Start y Warm-Start del Test 2	11
12.	Trazas de Cold-Start y Warm-Start del Test 3	11

Instrucciones

El presente documento corresponde a una plantilla que incluye las informaciones que deben ser proveídas para evaluar la entrega.

La entrega se basa mayormente en capturas de pantalla de la consola, la mayoría de ellas vistas en clase más algunas acciones adicionales que deben ser descubiertas por cada estudiante.

Todos los textos en rojo a lo largo de la plantilla, junto con esta página de instrucciones, deben ser eliminadas antes de la compilación final que debe ser entregada por Moodle.

1. Crear función lambda

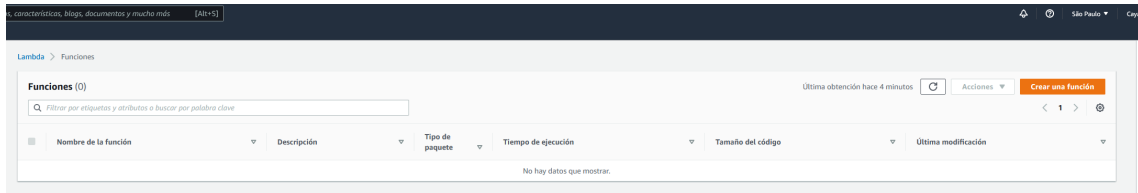


Figura 1: Captura de pantalla de la “AWS Management Console” posicionada en el servicio Lambda vista de Funciones

Seleccione una de las siguientes opciones para crear la función.

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

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

Información básica

Nombre de la función
Escriba un nombre para describir el propósito de la función.

Utilice exclusivamente letras, números, guiones o guiones bajos. No incluya espacios.

Tiempo de ejecución [Info](#)
Elija el lenguaje que desea utilizar para escribir la función. Tenga en cuenta que el editor de código de la consola solo admite Node.js, Python y Ruby.

Permisos [Info](#)
De forma predeterminada, Lambda creará un rol de ejecución con permisos para cargar registros en Amazon CloudWatch Logs. Puede personalizar este rol predeterminado más adelante al agregar los disparadores.

▼ Cambiar el rol de ejecución predeterminado

Rol de ejecución
Seleccione un rol que defina los permisos de la función. Para crear un rol personalizado, vaya a la [consola de IAM](#).

- ☒ Creación de un nuevo rol con permisos básicos de Lambda
- ☐ Uso de un rol existente
- ☐ Creación de un nuevo rol desde la política de AWS templates

ⓘ La creación de roles puede tardar unos minutos. No elimine el rol ni edite las políticas de confianza o de permisos de este rol.

Lambda creará un rol de ejecución denominado mifuncion-role-5m96oil, con permiso para cargar registros a Amazon CloudWatch Logs.

Figura 2: Captura de pantalla de “Create function” antes de crear la función

2. Configuración

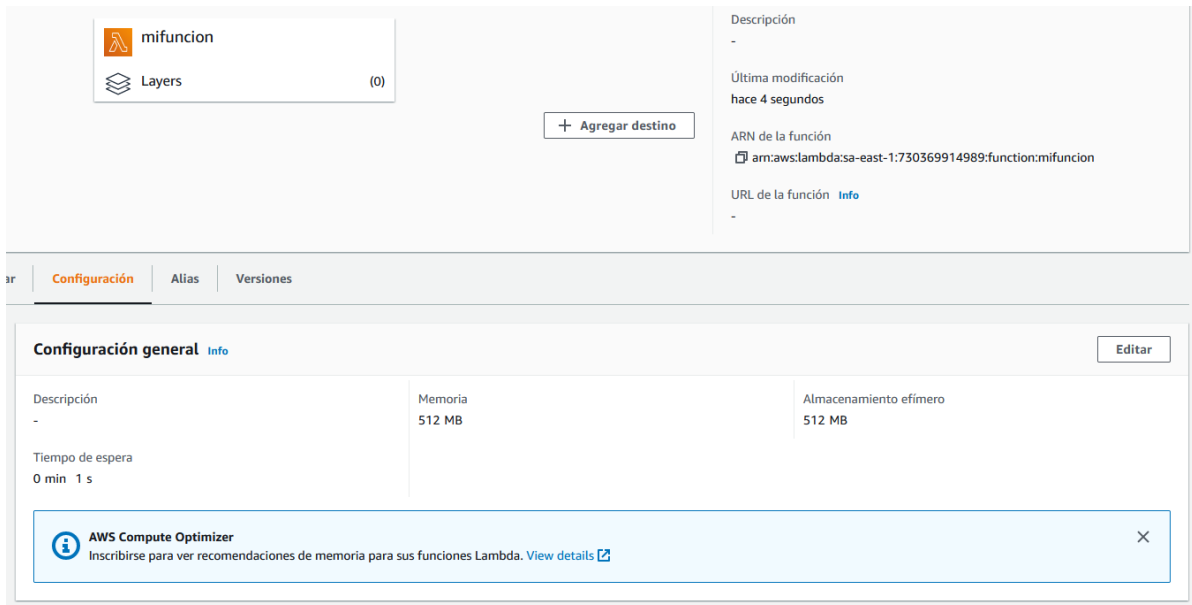


Figura 3: Captura de pantalla mostrando la configuración de la función a 512[MB] de memoria y un timeout de 1[s]

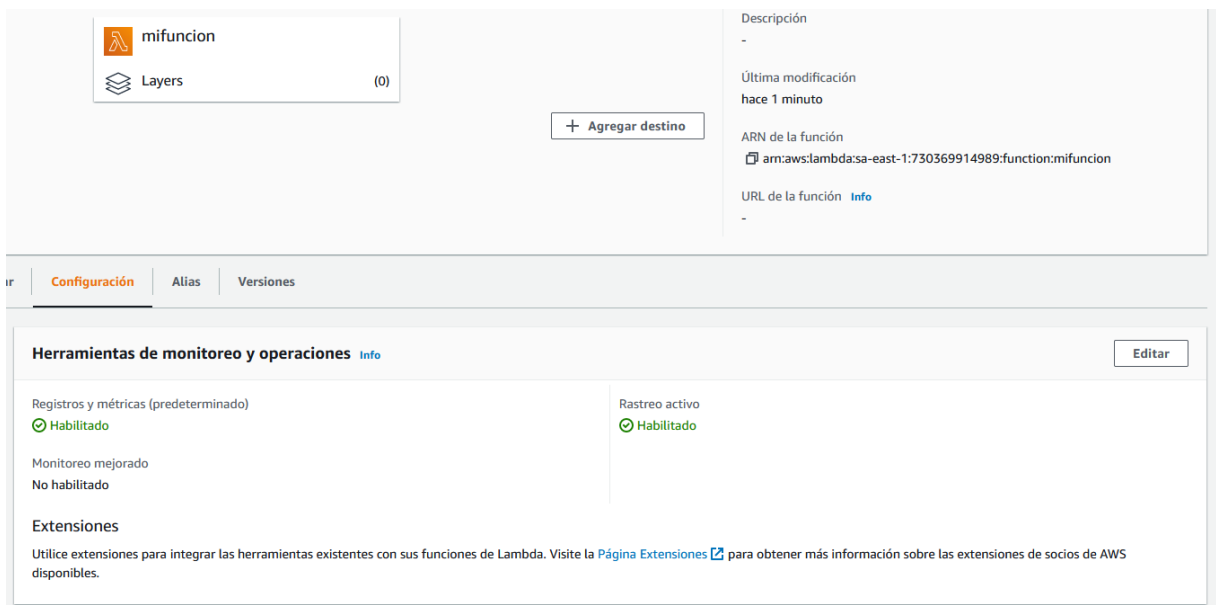


Figura 4: Captura de pantalla mostrando la habilitación de tracing con X-Ray

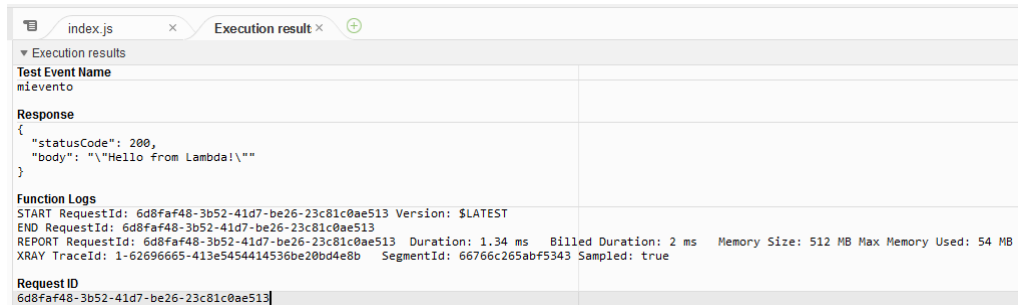


Figura 5: Captura de pantalla mostrando el resultado de la ejecución del test por omisión en cold-start (Duración facturada 2[ms], duración Cold-Start 1.34[ms] y Memoria usada 54[MB])

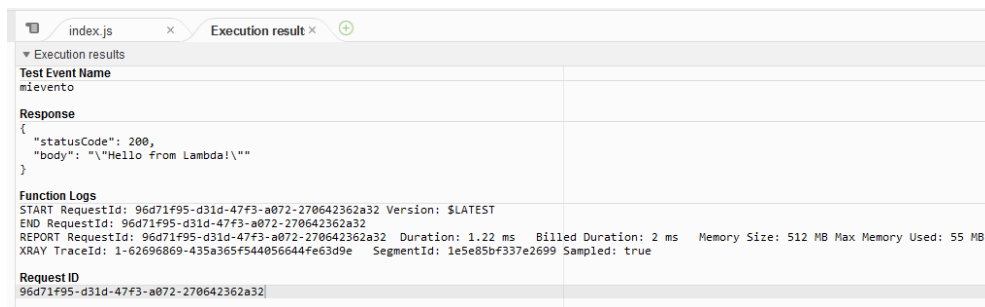


Figura 6: Captura de pantalla mostrando el resultado de la ejecución del test por omisión en warm-start (Duración facturada 2[ms] y Memoria usada 55[MB])

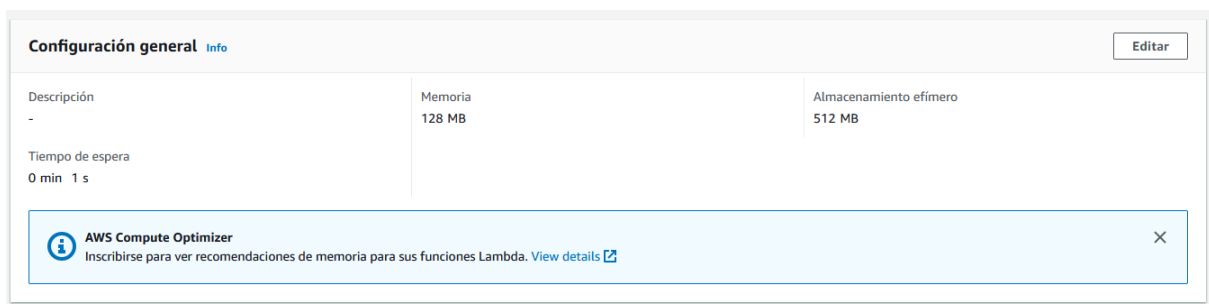


Figura 7: Captura de pantalla mostrando la re-configuración de la función a 128[MB] de memoria y un timeout de 1[s] optimo para el testeo de la ejecución ejemplo

3. Implementación

Evento de prueba

Para invocar la función sin guardar un ev

Acción de evento de prueba

☐ Crear un nuevo evento

Nombre del evento

Evento1

Evento JSON

```
1 {  
2   "protocol": "http",  
3   "host": "inf.utfsm.cl",  
4   "path": "/index.html"  
5 }
```

Evento de prueba

Para invocar la función sin guardar un evento,

Acción de evento de prueba

☐ Crear un nuevo evento

Nombre del evento

Evento2

Evento JSON

```
1 {  
2   "protocol": "https",  
3   "host": "www.inf.utfsm.cl",  
4   "path": "/"  
5 }
```

Evento de prueba

Para invocar la función sin guardar un evento, modifique el evento y luego

Acción de evento de prueba

☐ Crear un nuevo evento

Nombre del evento

Evento3

Evento JSON

```
1 {  
2   "protocol": "https",  
3   "host": "it.wikisource.org",  
4   "path": "/wiki/Divina_Commedia/Inferno/Canto_I"  
5 }
```

Figura 8: Capturas de pantalla mostrando las 3 pruebas consideradas

El siguiente código muestra la función implementada, el cual toma los 3 parámetros del evento para realizar un request GET que permite extraer la página web indicada por host y path bajo el protocolo determinado.

```

1 const http = require('http');
2 //const https = require('https');
3
4 exports.handler = async (event) => {
5     const req = await http.get(`${event.protocol}://${event.host}${event.path}`);
6     //const req = await https.get(`${event.protocol}://${event.host}${event.path}`);
7     return JSON.stringify(req);
8 };

```

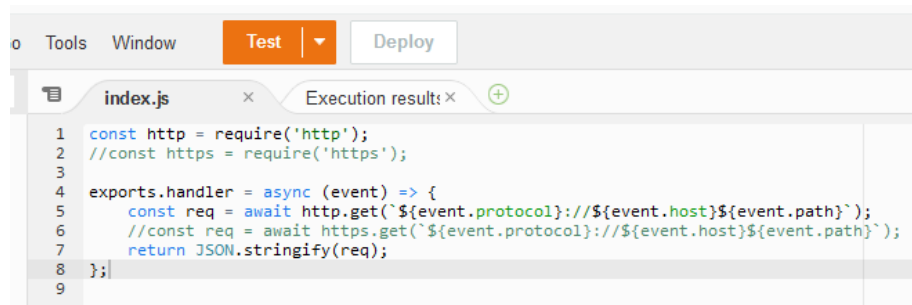


Figura 9: Captura de pantalla mostrando el código desplegado y listo para ejecutar en la consola de AWS

4. Ejecución

La ejecución de los diferentes test nos permite determinar las siguientes conclusiones sobre el contenido que se obtuvo desde el request:

- `http://inf.utfsm.cl/index.cl`: Se cambió el tiempo a 5s.
Por algún motivo desconocido el warm start me tiró error sin importar lo que hiciera. Le pregunté a un compañero que ya había hecho la tarea si le pasaba lo mismo (él no había tenido problema con el warm start anteriormente) y me respondió que sí, por lo que concluyo que es un problema de aws.
- `https://www.inf.utfsm.cl/index.cl`: Se cambió el tiempo a 5s.
Por algún motivo desconocido el warm start me tiró error sin importar lo que hiciera. Le pregunté a un compañero que ya había hecho la tarea si le pasaba lo mismo (él no había tenido problema con el warm start anteriormente) y me respondió que sí, por lo que concluyo que es un problema de aws.

- https://it.wikisource.org/wiki/Divina_Commedia/Inferno/Canto_I: Se cambió el tiempo a 5s.

Por algún motivo desconocido el warm start me tiró error sin importar lo que hiciera. Le pregunté a un compañero que ya había hecho la tarea si le pasaba lo mismo (él no había tenido problema con el warm start anteriormente) y me respondió que sí, por lo que concluyo que es un problema de aws.

La tabla 4 muestra los tiempos de ejecución medidos reportados por la consola

Test	Max. Mem. [MB]	Cold-Start [ms]	Ejec. [ms]
Test 1	55	155.77	50.12
Test 2	56	173.00	94.91
Test 3	56	158.60	90.14

Tabla 1: Mediciones en la ejecución de las diferentes pruebas realizadas

5. X-Ray

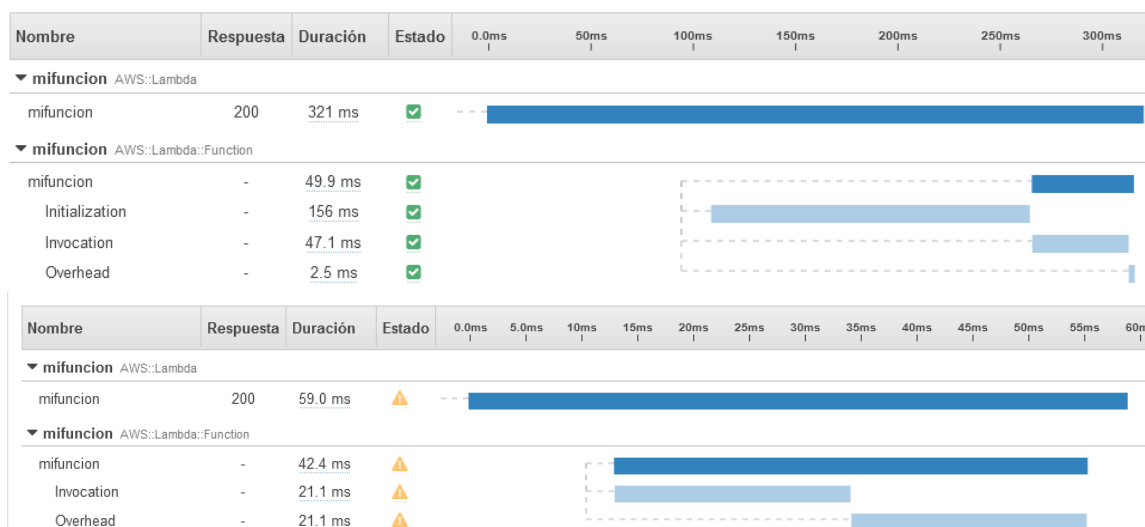


Figura 10: Trazas de Cold-Start y Warm-Start del Test 1

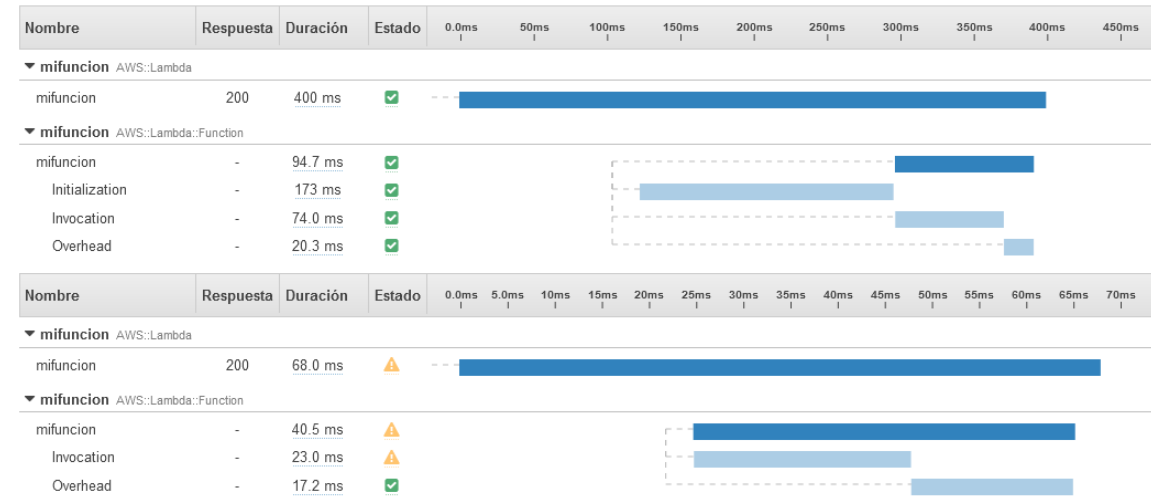


Figura 11: Trazas de Cold-Start y Warm-Start del Test 2

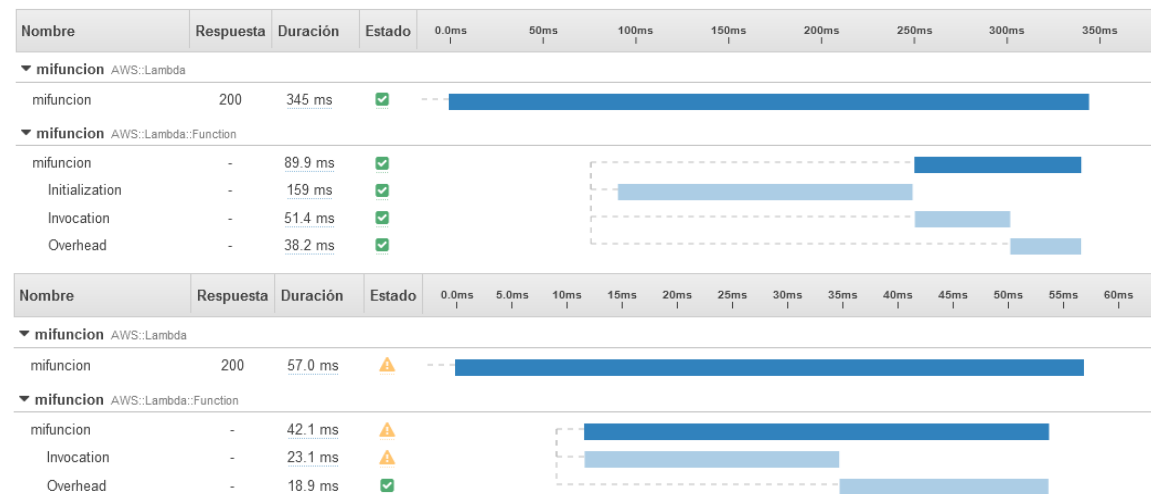


Figura 12: Trazas de Cold-Start y Warm-Start del Test 3