

CS2032 - Cloud Computing (Ciclo 2024-2)

Serverless

Semana 11 - Taller 3: Optimización de Lambdas

ELABORADO POR: GERALDO COLCHADO

- 1. Objetivo del taller 3
- 2. Ejercicio 1: Configuración de memoria RAM
- 3. Ejercicio 2: Stage o Etapa del Lambda/Api G.
- 4. Ejercicio 3: Ejecutar api rest desde Linux
- 5. Ejercicio 4: Ejercicio propuesto
- 6. Cierre

Optimización de Lambdas Objetivo del Taller 4

- Aprender a optimizar tiempos de ejecución de Lambdas
- Aprender a desplegar lambdas/Api G. en diferentes etapas
- Aprender a ejecutar un Api Rest desde Linux

- 1. Objetivo del taller 3
- 2. Ejercicio 1: Configuración de memoria RAM
- 3. Ejercicio 2: Stage o Etapa del Lambda/Api G.
- 4. Ejercicio 3: Ejecutar api rest desde Linux
- 5. Ejercicio 4: Ejercicio propuesto
- 6. Cierre

Paso 0: Crear los siguientes buckets en S3 de un supermercado. Suba las imágenes indicadas por el docente

catalogo-productos

- imagenes-chicas
- imagenes-medianas

documentos-ventas

- boletas
- facturas

documentos-compras

- ordenes-compra
- facturas
- guias-remision

Analice la memoria RAM asignada a cada lambda en serverless.yml

Paso 1: Implementar un lambda que liste todos los buckets y otro lambda que liste todos los objetos de un bucket. Ambos lambdas los debe publicar en un mismo API Rest en Api Gateway.

Adicionalmente debe automatizar el despliegue con serverless framework: \$ serverless deploy

Utilice el código fuente en python entregado por el docente. Nota: No olvide actualizar el archivo credentials, el org y el role

```
:~/lambdas/api-s3 $ serverless deploy

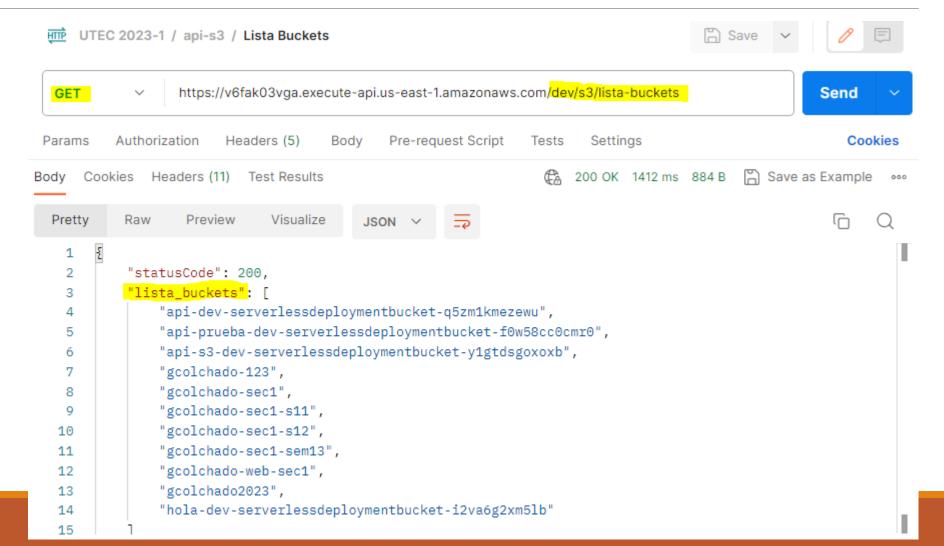
Deploying api-s3 to stage dev (us-east-1)

② Service deployed to stack api-s3-dev (84s)

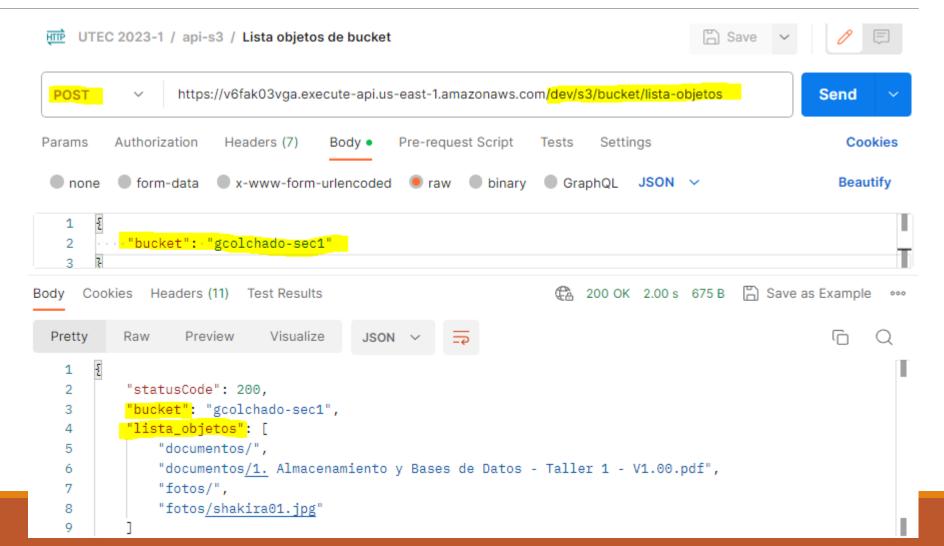
endpoints:
    GET - https://v6fak03vga.execute-api.us-east-1.amazonaws.com/dev/s3/lista-buckets
    POST - https://v6fak03vga.execute-api.us-east-1.amazonaws.com/dev/s3/bucket/lista-objetos
functions:
    lista_buckets: api-s3-dev-lista_buckets (746 B)
    lista_objetos_bucket: api-s3-dev-lista_objetos_bucket (746 B)
```

Need a faster logging experience than CloudWatch? Try our Dev Mode in Console: run "serverless dev"

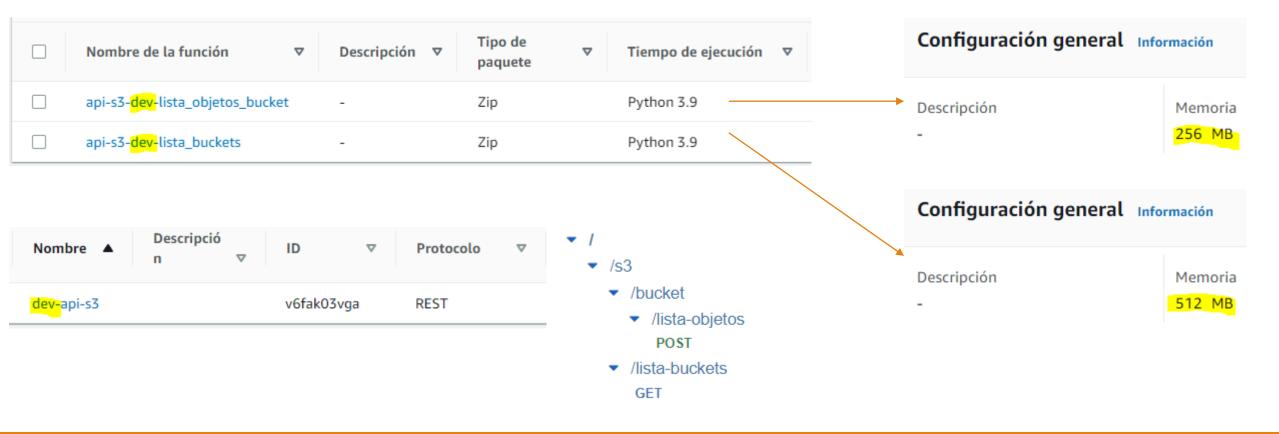
Paso 2: Pruebe en postman y valide y analice tiempo de ejecución



Paso 2: Pruebe en postman y valide y analice tiempo de ejecución



Paso 3: Analice lambda y api gateway creados



- 1. Objetivo del taller 3
- 2. Ejercicio 1: Configuración de memoria RAM
- 3. <u>Ejercicio 2: Stage o Etapa del Lambda/Api G.</u>
- 4. Ejercicio 3: Ejecutar api rest desde Linux
- 5. Ejercicio 4: Ejercicio propuesto
- 6. Cierre

Paso 1: La implementación del Api Rest "api-s3" en el stage "dev" (desarrollo) ha sido satisfactoria. Realice el despliegue automático en el entorno de pruebas.

\$ serverless deploy --stage test

```
:~/lambdas/api-s3 $ serverless deploy --stage test
Deploying api-s3 to stage test (us-east-1)
 Service deployed to stack api-s3-test (38s)
endpoints:
 GET - https://v1zxhd0j6d.execute-api.us-east-1.amazonaws.com/test/s3/lista-buckets
 POST - https://v1zxhd0j6d.execute-api.us-east-1.amazonaws.com/test/s3/bucket/lista-objetos
functions:
 lista_buckets: api-s3-test-lista_buckets (753 B)
 lista_objetos_bucket: api-s3-test-lista_objetos_bucket (753 B)
Improve API performance - monitor it with the Serverless Console: run "serverless --console"
```

Paso 2: Pruebe en postman

Paso 3: Analice lambda y api gateway creados. Note que son lambdas diferentes y el api gateway también es diferente al de stage = "dev"

Nombre de la función ▽	Descripción ▽	Tipo de paquete ▼	Tiempo de ejecución ▽	Última modificación ▽
api-s3- <mark>dev</mark> -lista_objetos_bucket	-	Zip	Python 3.9	hace 24 minutos
api-s3- <mark>test-</mark> lista_buckets	-	Zip	Python 3.9	hace 2 minutos
api-s3- <mark>dev-</mark> lista_buckets	-	Zip	Python 3.9	hace 24 minutos
api-s3- <mark>test</mark> -lista_objetos_bucket	-	Zip	Python 3.9	hace 2 minutos
Nombre Descripción	▽ ID 1	Protocolo		
dev-api-s3	v6fak03vga	REST		
test-api-s3	v1zxhd0j6d	REST		

Paso 4: La implementación del Api Rest "api-s3" en el stage "test" (pruebas) ha sido satisfactoria y el equipo de QA ha dado el OK. Realice el despliegue automático en el entorno de producción.

\$ serverless deploy --stage prod

```
:~/lambdas/api-s3 $ serverless deploy --stage prod
Deploying api-s3 to stage prod (us-east-1)
 Service deployed to stack api-s3-prod (90s)
endpoints:
 GET - https://m1n1z5u30d.execute-api.us-east-1.amazonaws.com/prod/s3/lista-buckets
  POST - https://m1n1z5u30d.execute-api.us-east-1.amazonaws.com/prod/s3/bucket/lista-objetos
functions:
  lista_buckets: api-s3-prod-lista_buckets (753 B)
  lista_objetos_bucket: api-s3-prod-lista_objetos_bucket (753 B)
Need a faster logging experience than CloudWatch? Try our Dev Mode in Console: run "serverless dev"
```

Paso 5: Pruebe en postman

Paso 6: Analice lambda y api gateway creados. Note que son lambdas diferentes y api gateway diferentes en los 3 stages.





- 1. Objetivo del taller 3
- 2. Ejercicio 1: Configuración de memoria RAM
- 3. Ejercicio 2: Stage o Etapa del Lambda/Api G.
- 4. Ejercicio 3: Ejecutar api rest desde Linux
- 5. Ejercicio 4: Ejercicio propuesto
- 6. Cierre

Optimización de Lambdas Ejercicio 3: Ejecutar api rest desde Linux

Paso 1: Ejecutar el api rest de Listar buckets (GET)

\$ curl https://v6fak03vga.execute-api.us-east-1.amazonaws.com/dev/s3/lista-buckets

```
:~/lambdas/api-s3 $ curl https://v6fak03vga.execute-api.us-east-1.amazonaws.com/dev/s3/lista-buckets
{"statusCode": 200, "lista_buckets": ["api-dev-serverlessdeploymentbucket-q5zm1kmezewu", "api-prueba-dev-serverlessdeploymentbucket-y1gtdsgoxoxb", "api-s3-prod-serverlessdeploymentbucket-y1gtdsgoxoxb", "api-s3-prod-serverlessdeploymentbucket-s4axnr9ac23n", "api-s3-test-serverlessdeploymentbucket-1djkjousbbski", "gcolchado-123", "gcolchado-sec1", "gcolchado-sec1", "gcolchado-sec1-s1", "gcolchado-sec1-s12", "gcolchado-sec1-sem13", "gcolchado-web-sec1", "gcolchado2023", "hola-dev-serverlessdeploymentbucket-i2va6g2xm5lb"]}:~/lambdas/api-s3 $
```

Optimización de Lambdas Ejercicio 3: Ejecutar api rest desde Linux

Paso 2: Ejecutar el api listar objetos de bucket (POST)

```
$ curl -X POST https://v6fak03vga.execute-api.us-east-1.amazonaws.com/dev/s3/bucket/lista-objetos - H 'Content-Type: application/json' -d '{"bucket":"gcolchado-sec1"}'
```

```
:~/lambdas/api-s3 $ curl -X POST https://v6fak03vga.execute-api.us-east-1.amazonaws.com/dev/s3/bucket/lista-obje
tos -H 'Content-Type: application/json' -d '{"bucket":"gcolchado-sec1"}'
{"statusCode": 200, "bucket": "gcolchado-sec1", "lista_objetos": ["documentos/", "documentos/1. Almacenamiento y
Bases de Datos - Taller 1 - V1.00.pdf", "fotos/", "fotos/shakira01.jpg"]}:~/lambdas/api-s3 $
```

- 1. Objetivo del taller 3
- 2. Ejercicio 1: Configuración de memoria RAM
- 3. Ejercicio 2: Stage o Etapa del Lambda/Api G.
- 4. Ejercicio 3: Ejecutar api rest desde Linux
- 5. Ejercicio 4: Ejercicio propuesto
- 6. Cierre

Optimización de Lambdas Ejercicio 4: Ejercicio propuesto (30 puntos)

- Modifique el serverless.yml para agregar 3 funciones:
 - Crear un nuevo bucket
 - Crear un nuevo directorio en un bucket existente
 - Subir un archivo (*) en un directorio de un bucket existente
- Realice el despliegue en el stage de dev, test y prod
- Pruebe el api rest desde Linux con curl.
- Muestre las evidencias de ejecución en linux y postman e imágenes de buckets, directorios y archivos
- Coloque toda la evidencia en un archivo word, excel o power point y súbalo al padlet indicado por el profesor.

- 1. Objetivo del taller 3
- 2. Ejercicio 1: Configuración de memoria RAM
- 3. Ejercicio 2: Stage o Etapa del Lambda/Api G.
- 4. Ejercicio 3: Ejecutar api rest desde Linux
- 5. Ejercicio 4: Ejercicio propuesto
- 6. Cierre

Cierre: Optimización de Lambdas - Qué aprendimos?

- Optimizar tiempos de ejecución de Lambdas
- Desplegar lambdas/Api G. en diferentes etapas
- Ejecutar un Api Rest desde Linux

Gracias

Elaborado por docente: Geraldo Colchado