

CS2032 - Cloud Computing (Ciclo 2024-2) Monitoreo de Infraestructura y Aplicaciones Semana 12 - Taller 1: CloudWatch (Métricas y Logs)

ELABORADO POR: GERALDO COLCHADO

- Objetivo del taller 1
- 2. Ejercicio 1: Registrar Logs en api-pelicula
- 3. Ejercicio 2: Ver Logs en CloudWatch
- 4. Ejercicio 3: Queries con Logs Insights Métricas
- 5. Ejercicio 4: Queries con Logs Insights Logs
- 6. Ejercicio 5: Queries con Logs Insights Errores
- 7. Ejercicio 6: Ejercicio propuesto
- 8. Cierre

CloudWatch (Métricas y Logs) Objetivo del Taller 1

- Aprender a registrar Logs en los Lambdas
- Aprender a revisar Logs de ejecución de Lambdas en CloudWatch (Ejecuciones correctas y errores)
- Aprender a consultar Métricas de Lambdas
- Aprender a consultar Logs de Lambdas

- 1. Objetivo del taller 1
- 2. Ejercicio 1: Registrar Logs en api-pelicula
- 3. Ejercicio 2: Ver Logs en CloudWatch
- 4. Ejercicio 3: Queries con Logs Insights Métricas
- 5. Ejercicio 4: Queries con Logs Insights Logs
- 6. Ejercicio 5: Queries con Logs Insights Errores
- 7. Ejercicio 6: Ejercicio propuesto
- 8. Cierre

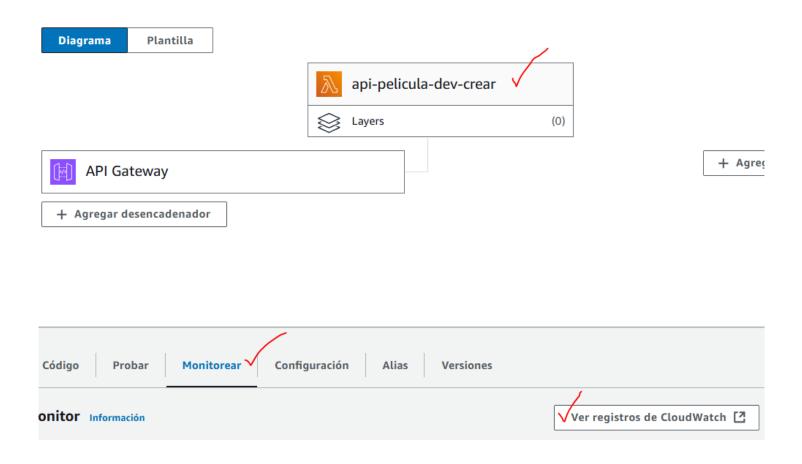
Ejercicio 1 Registrar Logs en api-pelicula

- **Paso 1:** Analice cómo se imprime en pantalla Logs en formato json
- **Paso 2:** Cree un repositorio en github api-pelicula y suba los archivos indicados por el docente
- **Paso 3:** En MV Serverless descargue con git clone, modifique en serverless.yml el **org** y **role** y actualice el **credentials**.
- Paso 4: Despliegue con \$ sls deploy
- Paso 5: Ejecute en Postman e inserte las 7 películas

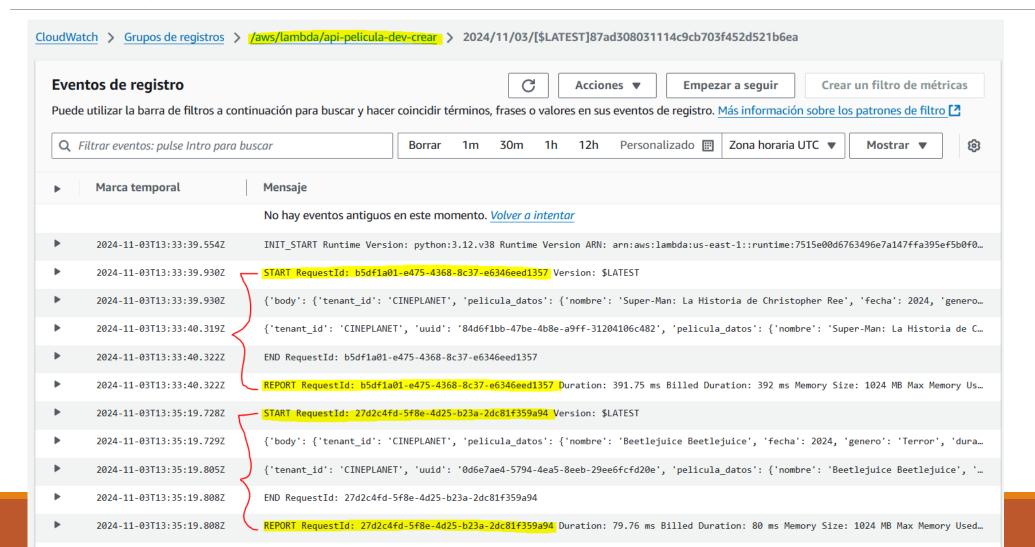
```
lambda handler(event, context):
# Entrada (json)
print(event) # Log json en CloudWatch
tenant_id = event['body']['tenant_id']
pelicula datos = event['body']['pelicula datos'
nombre tabla = os.environ["TABLE NAME"]
# Proceso
uuidv4 = str(uuid.uuid4())
pelicula = {
    'tenant id': tenant id,
    'uuid': uuidv4,
    'pelicula datos': pelicula datos
dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table(nombre tabla)
response = table.put item(Item=pelicula)
# Salida (json)
print(pelicula) # Log json en CloudWatch
return {
    'statusCode': 200,
    'pelicula': pelicula,
    'response': response
```

- 1. Objetivo del taller 1
- 2. Ejercicio 1: Registrar Logs en api-pelicula
- 3. Ejercicio 2: Ver Logs en CloudWatch
- 4. Ejercicio 3: Queries con Logs Insights Métricas
- 5. Ejercicio 4: Queries con Logs Insights Logs
- 6. Ejercicio 5: Queries con Logs Insights Errores
- 7. Ejercicio 6: Ejercicio propuesto
- 8. Cierre

Ejercicio 2 Ver Logs en CloudWatch

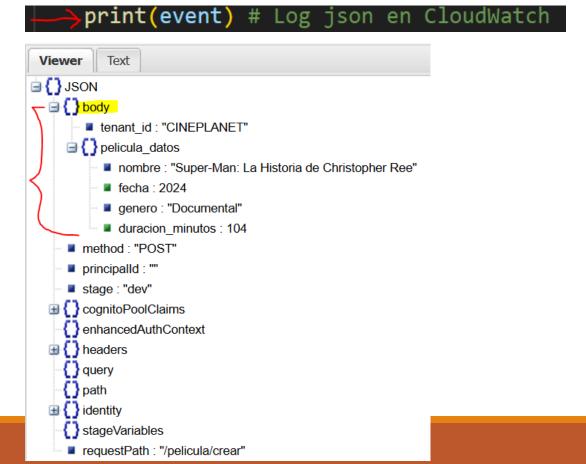


Ejercicio 2 Ver Logs en CloudWatch

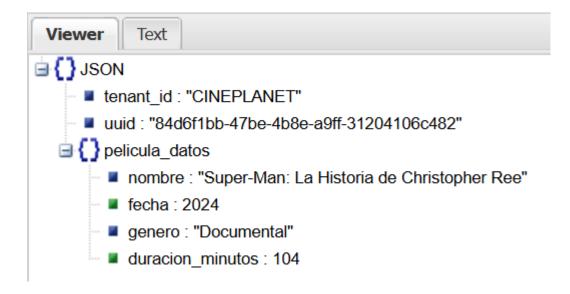


Ejercicio 2 Ver Logs en CloudWatch

Visualice los logs json en esta herramienta: https://jsonviewer.stack.hu/



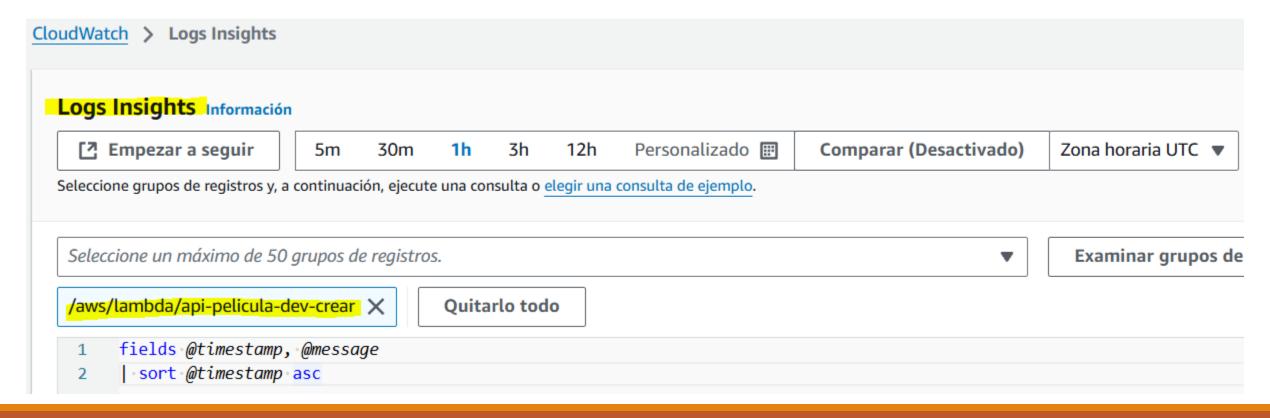
print(pelicula) # Log json en CloudWatch



- 1. Objetivo del taller 1
- 2. Ejercicio 1: Registrar Logs en api-pelicula
- 3. Ejercicio 2: Ver Logs en CloudWatch
- 4. Ejercicio 3: Queries con Logs Insights Métricas
- 5. Ejercicio 4: Queries con Logs Insights Logs
- 6. Ejercicio 5: Queries con Logs Insights Errores
- 7. Ejercicio 6: Ejercicio propuesto
- 8. Cierre

Ejercicio 3 Queries con Logs Insights - Métricas

fields @timestamp, @message | sort @timestamp asc



Ejercicio 3 Queries con Logs Insights - Métricas

fields @timestamp, @message, @billedDuration, @memorySize, @requestId | sort @timestamp asc | filter @type = "REPORT"

#	@timestamp	@message	@billedDuration	@memorySize	@requestId
> 1	2024-11-03T13:33:40.322Z	REPORT RequestId: b5df1	392.0	1.024E9	b5df1a01-e475-4368-8c37-e6346eed1357
> 2	2024-11-03T13:35:19.808Z	REPORT RequestId: 27d2c	80.0	1.024E9	27d2c4fd-5f8e-4d25-b23a-2dc81f359a94
> 3	2024-11-03T13:35:32.886Z	REPORT RequestId: 1da2f	81.0	1.024E9	1da2f1b3-1a30-4d0f-b075-77afa6aa2020
> 4	2024-11-03T13:35:47.610Z	REPORT RequestId: 15b3f	88.0	1.024E9	15b3f384-5318-4083-a907-b61a4e0a2b5e
> 5	2024-11-03T13:35:58.695Z	REPORT RequestId: 99f47	80.0	1.024E9	99f47537-c72f-491c-81f2-e1e54f4b0631
▶ 6	2024-11-03T13:36:10.507Z	REPORT RequestId: 49e34	78.0	1.024E9	49e34ffb-511e-4d06-b08f-dc2053e7e210
▶ 7	2024-11-03T14:17:16.615Z	REPORT RequestId: 4c69e	4.0	1.024E9	4c69eaf7-45b0-4f3a-a757-5c82a2a6b0a8

- 1. Objetivo del taller 1
- 2. Ejercicio 1: Registrar Logs en api-pelicula
- 3. Ejercicio 2: Ver Logs en CloudWatch
- 4. Ejercicio 3: Queries con Logs Insights Métricas
- 5. Ejercicio 4: Queries con Logs Insights Logs
- 6. Ejercicio 5: Queries con Logs Insights Errores
- 7. Ejercicio 6: Ejercicio propuesto
- 8. Cierre

fields @timestamp, @message, pelicula_datos.nombre, pelicula_datos.genero, pelicula_datos.fecha, pelicula_datos.duracion_minutos | sort @timestamp asc | filter tenant_id = "CINEPLANET"

#	@timestamp	@message	pelicula_datos.nombre	pelicula_datos.genero	pelicula_datos.fecha	pelicula_datos.duracion_minutos
) 1	2024-11-03T13:33:40.319Z	{'tenant_id	Super-Man: La Historia de Christopher Ree	Documental	2024	104
▶ 2	2024-11-03T13:35:19.805Z	{'tenant_id	Beetlejuice Beetlejuice	Terror	2024	110
▶ 3	2024-11-03T13:35:32.882Z	{'tenant_id	Godzilla Minus One Blanco y Negro	Accion	2023	130
▶ 4	2024-11-03T13:35:47.607Z	{'tenant_id	Dalia y el Libro Rojo	Animacion	2023	110
▶ 5	2024-11-03T13:35:58.692Z	{'tenant_id	El Apocalipsis de San Juan	Documental	2023	120
▶ 6	2024-11-03T13:36:10.504Z	{'tenant_id	La Sustancia	Terror	2024	140

fields @timestamp, @message, pelicula_datos.nombre, pelicula_datos.genero, pelicula_datos.fecha, pelicula_datos.duracion_minutos | sort pelicula_datos.duracion_minutos desc | filter tenant | id = "CINEPLANET"

#	@timestamp	@message	pelicula_datos.nombre	pelicula_datos.genero	pelicula_datos.fecha	pelicula_datos.duracion_minutos
) 1	2024-11-03T13:36:10.504Z	{'tenant_id	La Sustancia	Terror	2024	140
> 2	2024-11-03T13:35:32.882Z	{'tenant_id	Godzilla Minus One Blanco y Negro	Accion	2023	130
▶ 3	2024-11-03T13:35:58.692Z	{'tenant_id	El Apocalipsis de San Juan	Documental	2023	120
▶ 4	2024-11-03T13:35:19.805Z	{'tenant_id	Beetlejuice Beetlejuice	Terror	2024	110
▶ 5	2024-11-03T13:35:47.607Z	{'tenant_id	Dalia y el Libro Rojo	Animacion	2023	110
▶ 6	2024-11-03T13:33:40.319Z	{'tenant_id	Super-Man: La Historia de Christopher Ree	Documental	2024	104 V

fields @timestamp, @message, pelicula_datos.nombre, pelicula_datos.genero, pelicula_datos.fecha, pelicula_datos.duracion_minutos | sort pelicula_datos.duracion_minutos desc

filter tenant_id = "CINEPLANET" and pelicula_datos.genero = "Terror"

#	@timestamp	@message	pelicula_datos.nombre	pelicula_datos.genero	pelicula_datos.fecha	pelicula_datos.duracion_minutos
▶ 1	2024-11-03T13:36:10.504Z	{'tenant_id	La Sustancia	Terror	2024	140
> 2	2024-11-03T13:35:19.805Z	{'tenant_id	Beetlejuice Beetlejuice	Terror	2024	110

filter tenant_id = "CINEPLANET"
| stats count(*) by pelicula_datos.genero as genero

#	genero	count(*)
) 1	Terror	2
▶ 2	Documental	2
▶ 3	Animacion	1
• 4	Accion	1

- 1. Objetivo del taller 1
- 2. Ejercicio 1: Registrar Logs en api-pelicula
- 3. Ejercicio 2: Ver Logs en CloudWatch
- 4. Ejercicio 3: Queries con Logs Insights Métricas
- 5. Ejercicio 4: Queries con Logs Insights Logs
- 6. Ejercicio 5: Queries con Logs Insights Errores
- 7. Ejercicio 6: Ejercicio propuesto
- 8. Cierre

Ejercicio 5 Queries con Logs Insights - Errores

fields @timestamp, @message | sort @timestamp asc | filter @message like "ERROR"

#	@timestamp	@message	
) 1	2024-11-03T14:17:16.613Z	<pre>[ERROR] KeyError: 'pelicula_datos' Traceback (most recent call last):</pre>	File "/var/task/ <mark>CrearPelicula.py", line 9,</mark> in lambda_handler

- 1. Objetivo del taller 1
- 2. Ejercicio 1: Registrar Logs en api-pelicula
- 3. Ejercicio 2: Ver Logs en CloudWatch
- 4. Ejercicio 3: Queries con Logs Insights Métricas
- 5. Ejercicio 4: Queries con Logs Insights Logs
- 6. Ejercicio 5: Queries con Logs Insights Errores
- 7. Ejercicio 6: Ejercicio propuesto
- 8. Cierre

Ejercicio 6 Ejercicio propuesto (30 pts)

Modifique CrearPelicula.py:

- Agregue la estructura de manejo de errores try ... except
- Estandarice el **formato json de log** para ejecuciones correctas ("INFO") y errores ("ERROR"):

```
{
        "tipo" = "INFO" / "ERROR",
        "log_datos": { ... }
}
```

- Modique el print() para que imprima el formato json de log estándar para ejecuciones correctas ("INFO") y errores ("ERROR").
- Realice queries sobre el nuevo formato json de log estándar.
- Suba las evidencias al padlet.

- Objetivo del taller 1
- 2. Ejercicio 1: Registrar Logs en api-pelicula
- 3. Ejercicio 2: Ver Logs en CloudWatch
- 4. Ejercicio 3: Queries con Logs Insights Métricas
- 5. Ejercicio 4: Queries con Logs Insights Logs
- 6. Ejercicio 5: Queries con Logs Insights Errores
- 7. Ejercicio 6: Ejercicio propuesto
- 8. <u>Cierre</u>

Cierre: CloudWatch (Métricas y Logs) - Qué aprendimos?

- Aprender a registrar Logs en los Lambdas
- Aprender a revisar Logs de ejecución de Lambdas en CloudWatch (Ejecuciones correctas y errores)
- Aprender a consultar Métricas de Lambdas
- Aprender a consultar Logs de Lambdas

Gracias

Elaborado por docente: Geraldo Colchado