

Ejercicio programación sobre AWS

Repositorio de GITHUB: https://github.com/paulaarnaiz-lab/Practica_AWS_PaulaArnaiz.git

Credenciales activas (AWS Academy) y variables de entorno configuradas

Se exportan AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY, AWS_SESSION_TOKEN y AWS_DEFAULT_REGION para ejecutar la práctica.

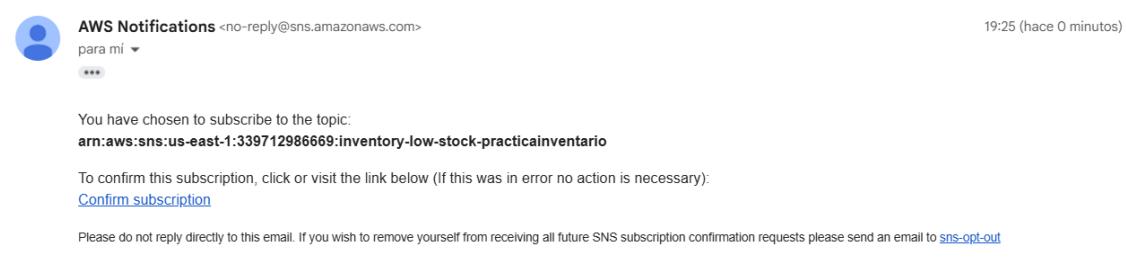
```
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS\infra> $env:AWS_ACCESS_KEY_ID="ASIAU6GDY7WY2L00EOSY"
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS\infra> $env:AWS_SECRET_ACCESS_KEY="G5Bmaba2u6gkGRMpxP7575MdMeNGD7ng79y5vkj"
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS\infra> $env:AWS_SESSION_TOKEN="I0cbJ3Jp22lu2xVEnI//////////w=CaTxVlxLd1c3QtMjGMEQCIGrrqwgui0
EN1Qy0e3+jnJmRaGB1KPBpYhrvlp9yA1AQzDBVHN82R/gmyHg058+SN=C7365x0PKP9B9X19hJQqvAgib//////////8BEAAADM0tOcxMj4N9jY2DSIMk/TdeE2gVh34ghgcMC1Pe/yiUh7Avsa
EN1481Jn-QFstxJdCu0xRis5+So1USNrxPg1Gy6lBuPwldt8UEg27p6ARQv6yh6cMoGioky7fPu8kBhZhd38gLvxF+EuIRxnpqBsbdU04ob0qTpvBLU2h0j50L4xdvF3ryu0z
ENw8huJzx6D0pDQcnHh0J_UxH7FtTS9+T51AE=116f10oN18c1Qj7fHnZC9pe18DFVH7398Hx10970rIgkndFyLb+165jUOpPaPQXG6zbzetzq/yqtAX991YATYuca9jZZReN1/GhXGe1C
A0+4Y3Ctqz298vgMvZcR5bnL8JgeaWYALfK-FeCkjfnwi3S1WDfT/HyrAr0zKOTWFOQuBnwInZNieltFcwAsIc1U9AzKpRtnLk5qxVpCabbfYsWfhFBmiPjN9EsQuSURPRS3Mu3mzjy3sW5wpCw
BB64dsCanJ_7DZPDU1tZyRKL-2Px2hvsyRgfTCGFFAiwbXvS6qbG91Dn068kgRpk/tguJ50G"
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS\infra> $env:AWS_DEFAULT_REGION="us-east-1"
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS\infra> aws sts get-caller-identity
{
    "UserId": "AROA6GDY7WY2L00EOSY",
    "Account": "339712986669",
    "Arn": "arn:aws:sts::339712986669:assumed-role/voclabs/user4417924=Paula_Arnaiz"
```

Deploy: recursos creados y endpoints generados

Ejecución de deploy.py creando buckets (uploads/web), tabla DynamoDB con streams, topic SNS, lambdas y API Gateway; se muestra la URL de la web y el endpoint /items.

Confirmación de suscripción SNS por email

Email de AWS SNS solicitando confirmación de suscripción al topic inventory-low-stock-....



Suscripción SNS confirmada

Página de AWS SNS mostrando “Subscription confirmed!” y el SubscriptionArn asociado al topic.

Subida de CSV a S3 (uploads bucket)

Carga del fichero inventario.csv al bucket de subidas inventory-uploads-practicainventario (trigger de ingestión para poblar DynamoDB).

```
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS> aws s3 cp .\inventario.csv s3://inventory-uploads-practicainventario/
upload: .\inventario.csv to s3://inventory-uploads-practicainventario/inventario.csv
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS> |
```

Respuesta API – GET /items (todo el inventario)

Respuesta de la API Gateway GET /items devolviendo el inventario completo en formato JSON.

Respuesta API – GET /items/Madrid (filtro por tienda)

Respuesta de la API Gateway GET /items/Madrid devolviendo el inventario filtrado por la tienda 'Madrid'.

The screenshot shows a browser window with the URL `qo1mcuclub.execute-api.us-east-1.amazonaws.com/items/Madrid`. The page content displays a large JSON array representing food items from a store in Madrid. Each item object contains fields such as `store`, `item`, `count`, and `store`. The JSON is very long and truncated at the end.

```
[{"store": "Madrid", "item": "Aqua 1.5l", "count": 26}, {"store": "Madrid", "item": "Azucar 1kg", "count": 20}, {"store": "Madrid", "item": "Bebida de avena 1l", "count": 16}, {"store": "Madrid", "item": "Cafe en grano 1kg", "count": 35}, {"store": "Madrid", "item": "Chocolate caliente", "count": 8}, {"store": "Madrid", "item": "Croissant", "count": 29}, {"store": "Madrid", "item": "Galletas", "count": 21}, {"store": "Madrid", "item": "Harina 1kg", "count": 14}, {"store": "Madrid", "item": "Huevos docena", "count": 18}, {"store": "Madrid", "item": "Leche entera 1l", "count": 22}, {"store": "Madrid", "item": "Mantequilla 250g", "count": 9}, {"store": "Madrid", "item": "Muffin chocolate", "count": 15}, {"store": "Madrid", "item": "Pan de molde", "count": 12}, {"store": "Madrid", "item": "Servilletas", "count": 60}, {"store": "Madrid", "item": "Tazas vasos", "count": 75}, {"store": "Madrid", "item": "Te negro", "count": 12}, {"store": "Madrid", "item": "Te verde", "count": 18}, {"store": "Madrid", "item": "Vaso carton", "count": 75}, {"store": "Madrid", "item": "Zumo naranja 1l", "count": 11}]
```

Web S3 – inventario cargado (Cargar todo)

Web estática (S3) consumiendo la API y mostrando el inventario completo en tabla (botón ‘Cargar todo’).

Inventario			
Tienda:	<input type="text" value="p. ej. Berlin"/>	<button>Cargar todo</button>	<button>Cargar tienda</button>
Store	Item	Count	
Valencia	Agua 1.5L	20	
Valencia	Azucar 1kg	16	
Valencia	Bebida de avena 1L	13	
Valencia	Cafe en grano 1kg	27	
Valencia	Chocolate caliente	6	
Valencia	Croissant	24	
Valencia	Galletas	18	
Valencia	Harina 1kg	12	
Valencia	Huevos docena	15	
Valencia	Leche entera 1L	19	
Valencia	Leche sin lactosa 1L	8	
Valencia	Mantequilla 250g	7	
Valencia	Muffin chocolate	12	
Valencia	Pan de molde	10	
Valencia	Servilletas	45	
Valencia	Tapas vasos	62	

Web S3 – inventario cargado por tienda (Madrid)

Web estática (S3) consumiendo la API y mostrando el inventario filtrado por tienda ‘Madrid’ (botón ‘Cargar tienda’).

The screenshot shows a web browser window titled 'Inventario'. A search bar at the top has 'Tienda: Madrid' selected. Below it is a table with three columns: 'Store', 'Item', and 'Count'. The table lists various grocery items and their counts from a store in Madrid. The items include Agua 1.5L (26), Azucar 1kg (20), Bebida de avena 1L (16), Cafe en grano 1kg (35), Chocolate caliente (8), Croissant (29), Galletas (21), Harina 1kg (14), Huevos docena (18), Leche entera 1L (22), Leche sin lactosa 1L (10), Mantequilla 250g (9), Muffin chocolate (15), Pan de molde (12), Servilletas (60), and Tapas vasos (75). A message at the bottom right says 'Mostrando 20 fila(s)'.

Store	Item	Count
Madrid	Agua 1.5L	26
Madrid	Azucar 1kg	20
Madrid	Bebida de avena 1L	16
Madrid	Cafe en grano 1kg	35
Madrid	Chocolate caliente	8
Madrid	Croissant	29
Madrid	Galletas	21
Madrid	Harina 1kg	14
Madrid	Huevos docena	18
Madrid	Leche entera 1L	22
Madrid	Leche sin lactosa 1L	10
Madrid	Mantequilla 250g	9
Madrid	Muffin chocolate	15
Madrid	Pan de molde	12
Madrid	Servilletas	60
Madrid	Tapas vasos	75

Forzar stock bajo (comando update-item)

Actualización manual en DynamoDB para forzar stock bajo (Count=0) y disparar el flujo DynamoDB Streams → Lambda → SNS.

```
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS> aws dynamodb update-item --cli-input-json file://item-low.json
```

Email recibido de SNS (Low stock)

Notificación recibida por email desde SNS (‘Low stock’) tras detectar stock por debajo del umbral configurado.

The email subject is 'Low stock: Berlin - StockPrueba'. It's from 'AWS Notifications <no-reply@sns.amazonaws.com>' and was sent '20:35 (hace 1 minuto)'. The message body contains a JSON object: {"type": "LOW_STOCK", "store": "Berlin", "item": "StockPrueba", "count": 0, "threshold": 2}. It includes a link to unsubscribe: <https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:339712986659:inventory-low-stock-practicainventario:caae5ee1-712a-4619-aca1-ea47d7643d09&Endpoint=pamalzcuenca@gmail.com>. A note at the bottom says 'Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>'.

Teardown

El script teardown.py elimina los recursos creados (API Gateway, Lambdas, mapeo de streams, SNS, DynamoDB y buckets S3).

```
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS\infra> python .\teardown.py
== TEARDOWN ==
Deleted API: 793t28gco4
Deleted Lambda: load_inventory_practicainventario
Deleted Lambda: get_inventory_api_practicainventario
Deleted mapping: 98767326-fb1e-4d3d-b992-bb58b51c9737
Deleted Lambda: notify_low_stock_practicainventario
Deleted SNS topic: arn:aws:sns:us-east-1:339712986669:inventory-low-stock-practicainventario
Deleted DynamoDB table: Inventory
Deleted bucket: inventory-uploads-practicainventario
Deleted bucket: inventory-web-practicainventario
== DONE ==
```