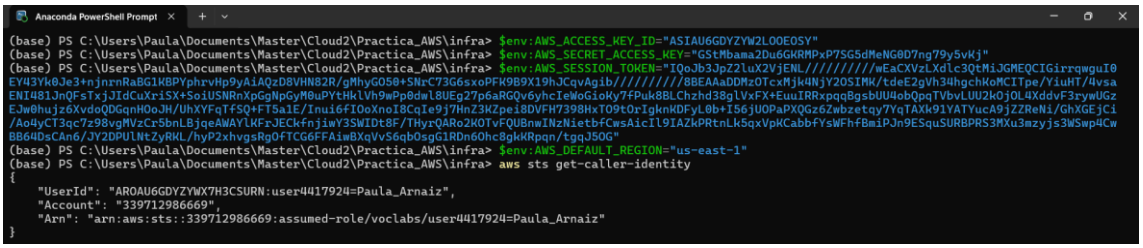
Paula Arnaiz Cuenca

Ejercicio programación sobre AWS

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

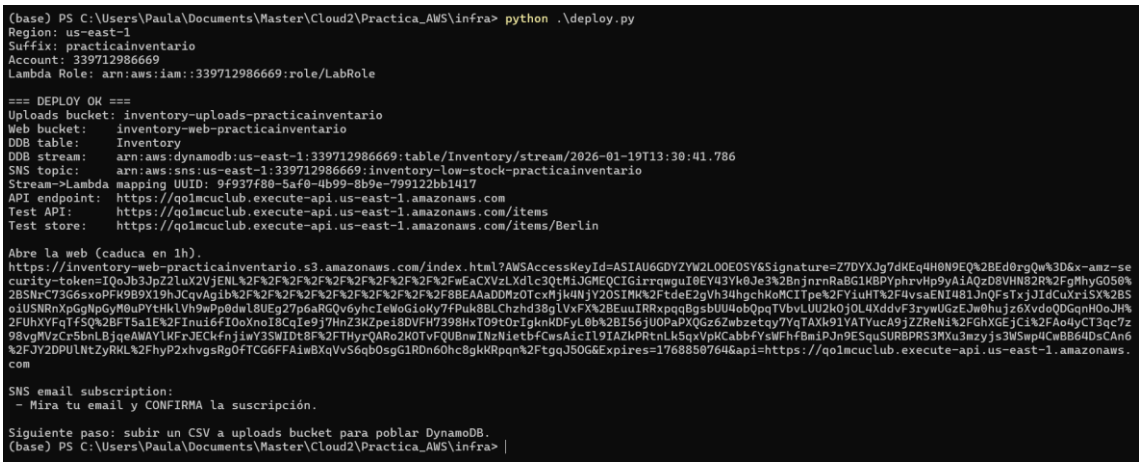
Credenciales activas (AWS Academy) y variables de entorno configuradas

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



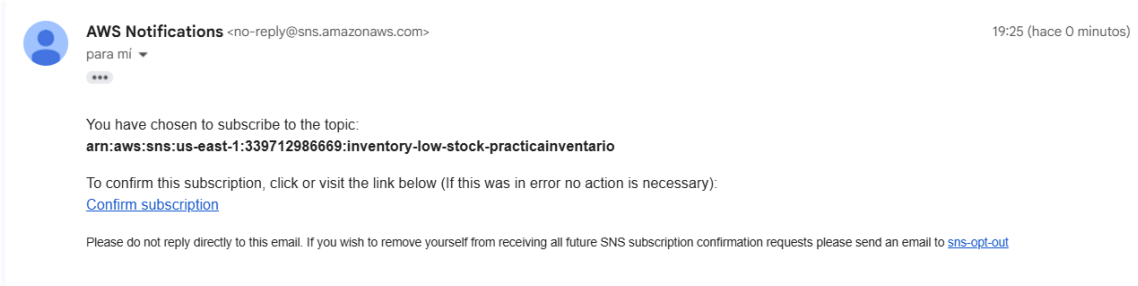
Deploy: recursos creados y endpoints generados

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.



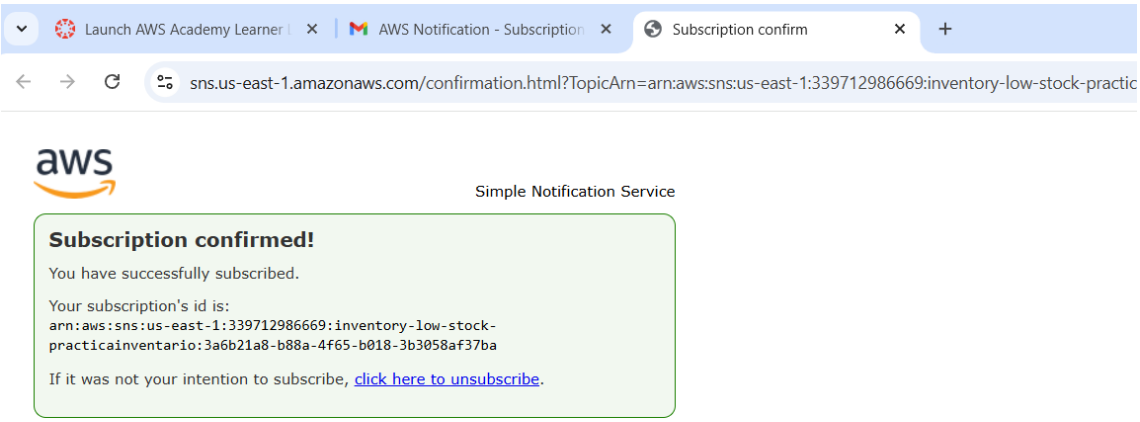
Confirmación de suscripción SNS por email

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 ingesta para poblar DynamoDB).

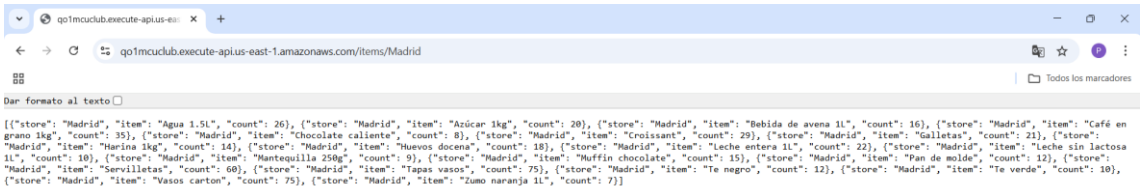
```
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS> aws s3 cp .\inventario.csv s3://inventory-uploads-practicainventario/
upload: .\inventario.csv to s3://inventory-uploads-practicainventario/inventario.csv
(base) PS C:\Users\Paula\Documents\Master\Cloud2\Practica_AWS>
```

Respuesta API – GET /items (todo el inventario)

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

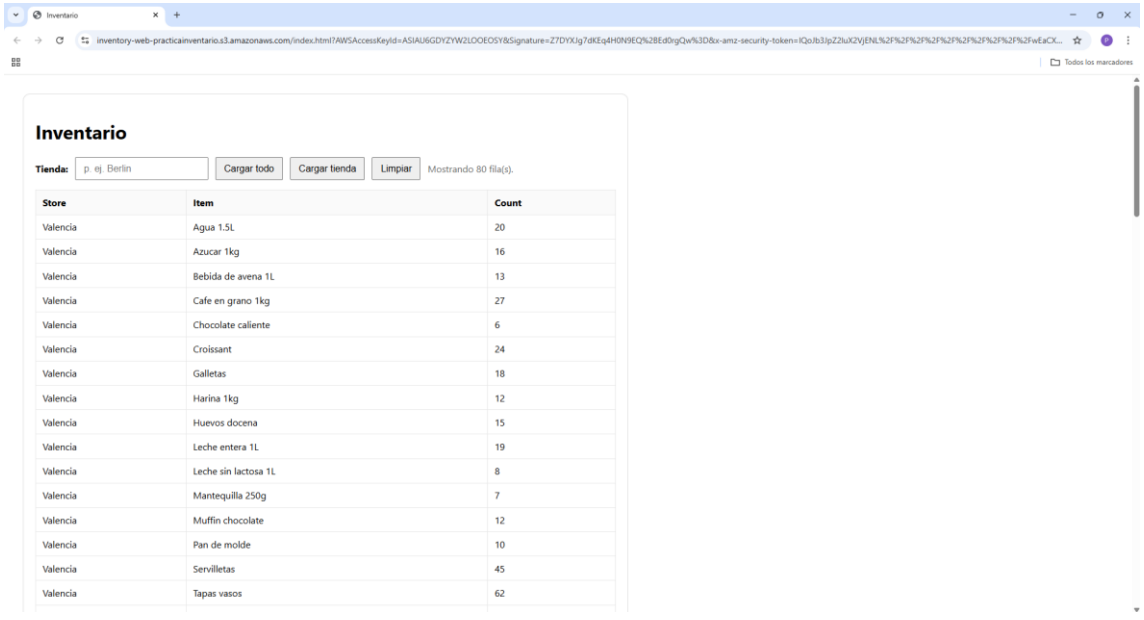
Respuesta API – GET /items/Madrid (filtro por tienda)

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



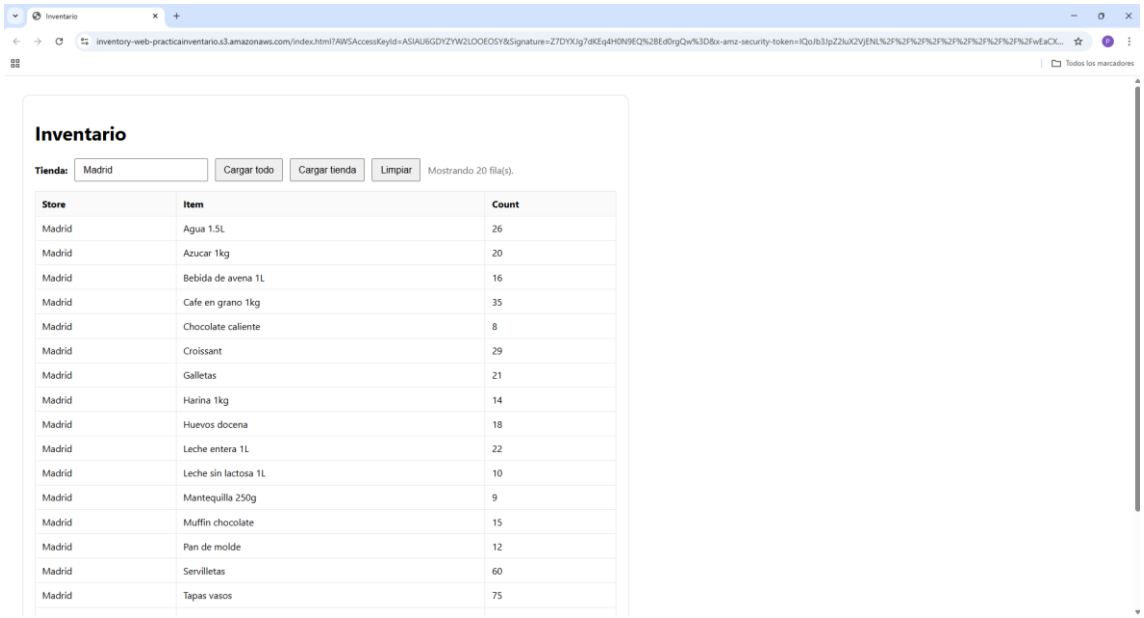
Web S3 – inventario cargado (Cargar todo)

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



Web S3 – inventario cargado por tienda (Madrid)

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



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

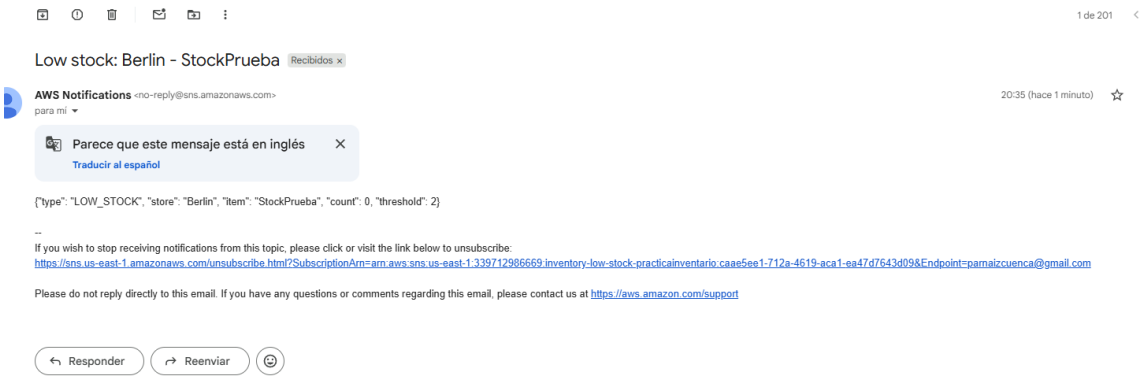
Forzar stock bajo (comando update-item)

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 fileb://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.



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 ==
```