

DOCUMENTACIÓN TÉCNICA Y DESPLIEGUE ECOMMERCE BOOKSTORE

Daniel Alejandro Castro Escobar

Laura Valentina Revelo Villareal

Katerine Valens Orejuela

Docente

Mario Germán Castillo Ramirez

Universidad ICESI

Ingeniería Telemática

2025

Proceso para desplegar la aplicación

Paso 1: Iniciar el entorno de sandbox

Se debe ingresar al sandbox, presionar “Start Lab” y esperar a que el ambiente esté listo.

Posterior a eso se ingresa a AWS.

Paso 2: Abrir cloud shell

En la parte superior derecha hay un icono donde se abre la consola de cloud shell. Esta se abre en la zona disponible.

Paso 3: Clonar el repositorio y ejecutar script de inicialización

Con el siguiente comando se clona el repositorio que se usará en este proyecto

```
git clone https://github.com/tobivalens/aws-ecommerce-infrastructure3.git
```

Luego se debe ingresar a la carpeta

```
cd ~/aws-ecommerce-infrastructure3/infrastructure
```

Se asignan los permisos de ejecución necesarios

```
chmod +x deploy.sh
```

Se ejecuta el script de inicialización

```
./deploy.sh
```

Paso 4: Abrir la dirección que arroja la salida del comando anterior en su navegador de preferencia

La salida puede ser algo como esto:

Tu aplicación está desplegada correctamente.

Accede aquí:

<http://bookstor-ALB-M9JX7wTfrV6p-2072659842.us-east-1.elb.amazonaws.com>

Explicación del funcionamiento del script “deploy.sh”

El script `deploy.sh` automatiza el proceso completo de despliegue de la infraestructura necesaria para la aplicación de comercio electrónico en AWS. Este archivo se encarga de preparar el entorno de ejecución generando un bucket S3 único para su uso durante la ejecución del proyecto y descargando automáticamente la plantilla donde se encuentra la infraestructura definida desde el repositorio remoto. Posteriormente, ejecuta el comando `aws cloudformation deploy` con los parámetros requeridos para levantar todos los recursos definidos como código. De esta manera, el script centraliza y simplifica todo el proceso de inicialización, garantizando un despliegue consistente en cada sesión del entorno sandbox.

El proceso interno cuando se ejecuta `deploy.sh` empieza cuando AWS procesa la plantilla de CloudFormation descargada y comienza a crear o actualizar los recursos definidos en ella. Internamente, CloudFormation inicia la construcción de la red, configura los grupos de seguridad y crea la base de datos RDS en las subredes privadas. Luego, genera un Launch Template con un script de inicialización que instala Node.js, Nginx y PM2, clona el repositorio de la aplicación y despliega automáticamente el backend y el frontend en las instancias gestionadas por el Auto Scaling Group. Finalmente, se implementa un Application Load Balancer, se activan alarmas de CloudWatch y se habilitan auditorías con CloudTrail.

Pruebas de funcionamiento

A continuación se describen las pruebas realizadas para validar el correcto funcionamiento de la infraestructura y de la aplicación desplegada en AWS. Estas pruebas abarcan desde la capa

de red hasta los componentes de cómputo, balanceo de carga, base de datos, monitoreo y auditoría.

7.1 Pruebas de red: VPC, subredes, tablas de ruteo, Internet Gateway y NAT Gateway

Para validar la capa de red se revisó, desde la consola de AWS, la correcta creación de la VPC del proyecto y de las cuatro subredes definidas en la plantilla de CloudFormation, correspondientes a dos subredes públicas y dos privadas. Se comprobó que los bloques CIDR de cada subred coincidieran con los parámetros definidos en el template , garantizando que no hubiera solapamientos de direcciones IP.

```
[ec2-user@ip-10-0-12-59 ~]$ curl https://www.google.com -I
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
content-security-policy-report-only: object-src 'none';base-uri 'self';script-src 'nonce-vw3MLUfEFsK-M2MdhUfQzg' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https; http:report-uri https://csp.withgoogle.com/csp/gws/other-hp
accept-ch: Sec-CH-Preferences-Color-Scheme
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Fri, 14 Nov 2025 20:30:31 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Fri, 14 Nov 2025 20:30:31 GMT
cache-control: private
set-cookie: AEC=AaJmaStJhOc3oP_Oay6BclugIIRa3WgItr7ziDRoKmy29QLzszhZ5GHKX8g; expires=Wed, 13-May-2026 20:30:31 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=lax
set-cookie: NID=526=ffQqp2IFUlnSLihqaTU3gbgir7j_DOAOAHjRKR6OF487UzrQywwi9nFX43RdrvIJt9c1mLjboL_ZHVWUlsA66ahJSgN0CwE5sX-8NVaDgtREFOFEV3rs-CAhW4EX6-L8o01PIgQqlgD-WRMymtUJLj-DoijXAJ_m98k7isKilEr6xKVHxx4tMLhCWgT35q4M3so8fv8TDMH-A6vRV6KC; expires=Sat, 16-May-2026 20:30:31 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
```

Your VPCs (2) Info

Last updated less than a minute ago Actions

Find VPCs by attribute or tag

	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 C
<input type="checkbox"/>	bookstore-vpc	vpc-0059ebbd0ffe3d1e3	Available	Off	10.0.0.0/16	-

Subnets (10) Info

Last updated 1 minute ago

Find subnets by attribute or tag

	Name	Subnet ID	State	VPC
<input type="checkbox"/>	bookstore-public-b	subnet-0adff4acdaf2de792	Available	vpc-0059ebbd0ffe3d1e3 bookstore-vpc
<input type="checkbox"/>	-	subnet-0856f1b0c3588d219	Available	vpc-0ae4cdccf79dcc929
<input type="checkbox"/>	-	subnet-0f727be1abff570d9	Available	vpc-0ae4cdccf79dcc929
<input type="checkbox"/>	-	subnet-0cc51273540878bb1	Available	vpc-0ae4cdccf79dcc929
<input type="checkbox"/>	bookstore-public-a	subnet-07e66c6b3ebf02c96	Available	vpc-0059ebbd0ffe3d1e3 bookstore-vpc
<input type="checkbox"/>	-	subnet-018a3fb6918edbcfa	Available	vpc-0ae4cdccf79dcc929
<input type="checkbox"/>	-	subnet-0698053578607a478	Available	vpc-0ae4cdccf79dcc929
<input type="checkbox"/>	bookstore-private-a	subnet-0471d2c7aaff16ff8	Available	vpc-0059ebbd0ffe3d1e3 bookstore-vpc
<input type="checkbox"/>	bookstore-private-b	subnet-0590d73a867e80b1f	Available	vpc-0059ebbd0ffe3d1e3 bookstore-vpc
<input type="checkbox"/>	-	subnet-0cd6d46232e0456e4	Available	vpc-0ae4cdccf79dcc929

Route tables (1/4) [Info](#)

Find route tables by attribute or tag

<input type="checkbox"/>	Name	Route table ID	Explicit subnet associ
<input type="checkbox"/>	-	rtb-00cacbc506bab6924	-
<input type="checkbox"/>	bookstore-public-rt	rtb-01eca78077478d444	2 subnets
<input checked="" type="checkbox"/>	bookstore-private-rt	rtb-0963ce4ea74146e7d	2 subnets
<input type="checkbox"/>	-	rtb-0dad7bed9b78a0be5	-

Posteriormente se verificaron las tablas de ruteo asociadas. En la tabla de ruteo pública se confirmó la presencia de una ruta por defecto (0.0.0.0/0) apuntando al Internet Gateway y su asociación con las subredes públicas. En la tabla de ruteo privada se validó que la ruta por defecto estuviera dirigida hacia el NAT Gateway desplegado en una subred pública.

Por último, para comprobar la salida a Internet desde una subred privada, se accedió a una instancia de aplicación ubicada en dicha subred y se ejecutó un comando de prueba, por ejemplo: `curl https://aws.amazon.com`.

La respuesta del comando confirmó que el tráfico saliente desde las subredes privadas se enruta correctamente hacia Internet a través del NAT Gateway.

7.2 Pruebas de instancias EC2 y Bastion Host

En cuanto a las instancias EC2, se validó la creación del Bastion Host en una subred pública mediante la consola de EC2, verificando que contara con una dirección IP pública asignada.

Utilizando la clave configurada para el proyecto, se comprobó que era posible establecer una conexión SSH hacia el Bastion mediante el comando: `ssh -i <keypair>.pem`

`ec2-user@<IP_PÚBLICA_BASTION>`.

Una vez dentro del Bastion, se realizaron pruebas de conectividad hacia las instancias del Auto Scaling Group ubicadas en las subredes privadas, accediendo a ellas a través de sus

direcciones IP privadas. Esto permitió verificar que los grupos de seguridad estaban configurados para permitir tráfico SSH únicamente desde el Bastion y que la comunicación entre la capa pública y la capa privada funcionaba correctamente. Estas pruebas confirmaron que el Bastion se encuentra correctamente configurado como punto de acceso administrativo seguro hacia la infraestructura interna.

7.3 Pruebas del balanceador de carga (ALB) y del Target Group

Para el balanceador de carga de aplicación (ALB), se revisó en la consola de EC2, dentro de la sección de Target Groups, el estado de las instancias registradas en el grupo de destino asociado al ALB. Una vez desplegado y configurado correctamente el backend, las instancias comenzaron a aparecer en estado “healthy”, gracias al endpoint de salud /health expuesto por la aplicación. También se utilizó el DNS público del ALB, obtenido desde los outputs del stack de CloudFormation, para validar el enrutamiento del tráfico. Desde el sandbox y desde un entorno local se ejecutó el comando: `curl http://<AlbDNS>`.

Target group healthy

bookst-Targe-LJ2G7ZA4AKGC

Actions

Details

arn:aws:elasticloadbalancing:us-east-1:344804352841:targetgroup/bookst-Targe-LJ2G7ZA4AKGC/e15964d88e66b6d8

Target type

Instance

IP address type

IPv4

Protocol : Port

HTTP: 3000

Load balancer

bookstor-ALB-M9JX7wTfrV6p

Protocol version

HTTP1

VPC

vpc-0059ebbd0ffe3d1e3

1

Total targets

1

Healthy

0 Anomalous

0

Unhealthy

0

Unused

0

Initial

0

Draining

Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Curl a la dirección del loadbalancer DNS

```

[ec2-user@ip-10-0-12-59 backend]$ curl http://bookstor-ALB-M9JX7wTfrV6p-2072659842.us-east-1.elb.amazonaws.com
curl: (3) URL rejected: No host part in the URL
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Login - Librería</title>
  <link rel="stylesheet" href="css/styles.css">
  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
  <!-- Aquí se define API URL = "/api" -->
  <script src="js/config.js"></script>
</head>
<body>
  <header>
    <h1>📖 Librería Online</h1>
  </header>

  <main>
    <section>
      <h2>Iniciar sesión</h2>
      <form id="form-login">
        <label>Usuario:</label><br>
        <input type="text" id="username" required><br><br>
        <label>Contraseña:</label><br>
        <input type="password" id="password" required><br><br>
        <button type="submit">Entrar</button>
      </form>
    </section>
  </main>
</body>
</html>

```

Cuando la aplicación estuvo correctamente desplegada, este comando devolvió una respuesta HTTP válida de la aplicación web, evidenciando que el ALB recibía el tráfico de los clientes y lo distribuía de forma correcta hacia las instancias de la capa de aplicación.

7.4 Pruebas de la base de datos RDS

Se verificó la correcta configuración del endpoint de la base de datos RDS en el archivo de variables de entorno (.env) del backend, asegurando que el host, nombre de base de datos, usuario y contraseña coincidieran con los parámetros definidos en la plantilla de infraestructura. Tras iniciar la aplicación, se revisaron los logs de Node.js y se observaron mensajes de conexión exitosa y sincronización de tablas, lo cual confirmó que la aplicación era capaz de conectarse y operar sobre la base de datos.

Como validación adicional, desde una instancia de aplicación se probó la conectividad directa hacia el puerto 3306 del RDS utilizando un comando como el siguiente: *nc -zv*

```
<DbEndpoint> 3306
```

La respuesta sin errores confirmó que el grupo de seguridad de la base de datos permitía conexiones únicamente desde el grupo de seguridad asociado a la aplicación, cumpliendo así las restricciones de acceso definidas en el diseño de seguridad.

Se muestran los datos almacenados en la base de datos

```
ec2-user@ip-10-0-12-59 ~]$ curl -v http://127.0.0.1:3000/api/products
* Trying 127.0.0.1:3000...
* Connected to 127.0.0.1 (127.0.0.1) port 3000
* using HTTP/1.x
> GET /api/products HTTP/1.1
> Host: 127.0.0.1:3000
> User-Agent: curl/8.11.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Type: application/json; charset=utf-8
< Content-Length: 1578
< ETag: W/"62a-40ZTnJqYiOPfufZJmk8ybf/IyBE"
< Date: Fri, 14 Nov 2025 21:21:16 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
[[{"id":1,"nombre":"El Principito","descripcion":"Cl\u00e1sico de Antoine de Saint-Exup\u00e9ry","precio":35000,"imagen":"img/principito.jpeg","createdAt":"2025-11-14T18:44:02.000Z","updatedAt":"2025-11-14T18:44:02.000Z"}, {"id":2,"nombre":"Cien a\u00f1os de soledad","descripcion":"Gabriel Garc\u00eda M\u00e1rquez","precio":58000,"imagen":"img/cienanosdesoledad.jpg","createdAt":"2025-11-14T18:44:02.000Z","updatedAt":"2025-11-14T18:44:02.000Z"}, {"id":3,"nombre":"Don Quijote de la Mancha","descripcion":"Miguel de Cervantes","precio":42000,"imagen":"img/donquijote.jpg","createdAt":"2025-11-14T18:44:02.000Z","updatedAt":"2025-11-14T18:44:02.000Z"}, {"id":4,"nombre":"Rayuela","descripcion":"Julio Cort\u00e1zar","precio":39000,"imagen":"img/rayuelabook.jpeg","createdAt":"2025-11-14T18:44:02.000Z","updatedAt":"2025-11-14T18:44:02.000Z"}, {"id":5,"nombre":"Dracula","descripcion":"Bram Stoker","precio":60000,"imagen":"img/dracula.jpeg","createdAt":"2025-11-14T18:44:02.000Z","updatedAt":"2025-11-14T18:44:02.000Z"}, {"id":6,"nombre":"Cumbres Borrascosas","descripcion":"Emily Bront\u00e9","precio":32000,"imagen":"img/cumbres.jpeg","createdAt":"2025-11-14T18:44:02.000Z","updatedAt":"2025-11-14T18:44:02.000Z"}, {"id":7,"nombre":"El retrato de Dorian Gray","descripcion":"Oscar Wilde","precio":32000,"imagen":"img/dorian.jpeg","createdAt":"2025-11-14T18:44:02.000Z","updatedAt":"2025-11-14T18:44:02.000Z"}, {"id":8,"nombre":"El Conde de Montecristo","descripcion":"Alexandre Dumas","precio":50000,"imagen":"img/montecristo.jpeg","createdAt":"2025-11-14T18:44:02.000Z","updatedAt":"2025-11-14T18:44:02.000Z"}]]
```

```
MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| bookstore |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.011 sec)
```

```
Database changed
MySQL [bookstore]> SHOW TABLES;
+-----+
| Tables_in_bookstore |
+-----+
| Products |
| Users |
+-----+
2 rows in set (0.001 sec)
```

```
MySQL [bookstore]> SELECT * FROM Products;
```

id	nombre	descripcion	precio	imagen	createdAt	updatedAt
1	El Principito	Cl\u00e1sico de Antoine de Saint-Exup\u00e9ry	35000	img/principito.jpeg	2025-11-14 18:44:02	2025-11-14 18:44:02
2	Cien a\u00f1os de soledad	Gabriel Garc\u00eda M\u00e1rquez	58000	img/cienanosdesoledad.jpg	2025-11-14 18:44:02	2025-11-14 18:44:02
3	Don Quijote de la Mancha	Miguel de Cervantes	42000	img/donquijote.jpg	2025-11-14 18:44:02	2025-11-14 18:44:02
4	Rayuela	Julio Cort\u00e1zar	39000	img/rayuelabook.jpeg	2025-11-14 18:44:02	2025-11-14 18:44:02
5	Dracula	Bram Stoker	60000	img/dracula.jpeg	2025-11-14 18:44:02	2025-11-14 18:44:02
6	Cumbres Borrascosas	Emily Bront\u00e9	32000	img/cumbres.jpeg	2025-11-14 18:44:02	2025-11-14 18:44:02
7	El retrato de Dorian Gray	Oscar Wilde	32000	img/dorian.jpeg	2025-11-14 18:44:02	2025-11-14 18:44:02
8	El Conde de Montecristo	Alexandre Dumas	50000	img/montecristo.jpeg	2025-11-14 18:44:02	2025-11-14 18:44:02


```
MySQL [bookstore]> SELECT * FROM Users;
```

id	username	password	createdAt	updatedAt
1	1	\$2a\$10\$5vpWT//CB2hKalboQv0UoOg5XYYGJ80PpBgZ/PxV92JwSKRVwzxs	2025-11-14 18:47:50	2025-11-14 18:47:50
2	2	\$2a\$10\$gRJqCJkjzdGg3Q42wIkzjo5idQUFBiql7yd3AWs15kcNbnDb76DGG	2025-11-14 21:52:49	2025-11-14 21:52:49

```
2 rows in set (0.001 sec)
```

7.5 Pruebas del Auto Scaling Group

En relación con el Auto Scaling Group, se verificó mediante la consola de EC2 y con el uso de la CLI de AWS que el grupo estuviera creando el número de instancias definido en el parámetro DesiredCapacity. Para ello se utilizó un comando similar a:

```
aws autoscaling describe-auto-scaling-groups \
    --auto-scaling-group-names <AsgName> \
    --region us-east-1
```

La salida del comando permitió confirmar que el ASG mantenía el número esperado de instancias en ejecución. Además, se realizó una prueba de recuperación ante fallos forzando la terminación de una de las instancias pertenecientes al grupo. Se observó cómo el Auto Scaling Group lanzaba automáticamente una nueva instancia de reemplazo hasta volver a alcanzar el tamaño deseado. Esta prueba validó el correcto funcionamiento de los mecanismos de auto-recuperación y la contribución del ASG a la alta disponibilidad del servicio.

```
{...skipping...
  "AutoScalingGroups": [
    {
      "AutoScalingGroupName": "bookstore-main-ASG-v5hZrJGaGvLA",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-east-1:344884352841:autoScalingGroup:2c81d6cb-6934-46a0-ab8f-b4afb064f826:autoScalingGroupName/bookstore-main-ASG-v5hZrJGaGvLA",
      "LaunchTemplate": {
        "LaunchTemplateId": "lt-021c54451e57707a9",
        "LaunchTemplateName": "bookstore-lt",
        "Version": "1"
      },
      "MinSize": 1,
      "MaxSize": 2,
      "DesiredCapacity": 1,
      "DefaultCooldown": 300,
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b"
      ],
      "LoadBalancerNames": [],
      "TargetGroupARNs": [
        "arn:aws:elasticloadbalancing:us-east-1:344884352841:targetgroup/bookst-Targe-LJ2G7ZAAKGC/e15964d88e66b6d8"
      ],
      "HealthCheckType": "ELB",
      "HealthCheckGracePeriod": 120,
      "Instances": [

```

7.6 Pruebas del backend y del frontend de la aplicación

Para el backend desarrollado en Node.js, se probó el endpoint de salud expuesto por la aplicación mediante la ejecución de: `curl localhost:3000/health`.

La respuesta JSON obtenida, similar a: `{"status":"ok"}` confirmó que la API se encontraba levantada y respondiendo correctamente en el puerto configurado, además de evidenciar que la aplicación era capaz de conectarse a la base de datos sin errores.

En cuanto al frontend, una vez generada la versión de producción (mediante `npm run build`) y copiado el contenido al directorio `public` del backend, se configuró Nginx como reverse proxy escuchando en el puerto 80 y redirigiendo el tráfico hacia el backend en el puerto interno de la aplicación. Para validar que la interfaz estuviera siendo servida correctamente, se ejecutó: `curl localhost` y luego se accedió desde un navegador utilizando el DNS del ALB. De forma visual se comprobó la carga correcta de la página principal de la aplicación, la visualización del catálogo de productos, la funcionalidad del carrito de compras (agregar y visualizar productos) y la navegación hacia las pantallas de autenticación y de checkout, lo que confirmó el correcto funcionamiento de la capa de presentación.

Confirmar que las carpetas fueron extraídas correctamente:

```
[ec2-user@ip-10-0-12-59 ~]$ cd /opt/app/src/backend
ls
index.js  models  node_modules  package-lock.json  package.json  public  routes  seed.js
[ec2-user@ip-10-0-12-59 backend]$
```

El archivo `.env` se generó como se esperaba.

```
[ec2-user@ip-10-0-12-59 backend]$ cat .env
DB_HOST=bookstore-mysql.cpkopbn1zoqo.us-east-1.rds.amazonaws.com
DB_USER=admin
DB_PASS=Book1234!
DB_NAME=bookstore
DB_PORT=3306

PORT=3000
JWT_SECRET=mi_secreto_super_seguro
MP_ACCESS_TOKEN=APP_USR-808920290219188-110702-110355d6da30d72e55b45d7b82c08624-2973454666
MP_PUBLIC_KEY=APP_USR-541a0dcd-d269-4379-a988-623634c72992
[ec2-user@ip-10-0-12-59 backend]$
```

El punto `/health` del backend respondió correctamente

```
[ec2-user@ip-10-0-12-59 backend]$ curl localhost:3000/health
{"status":"ok"}[ec2-user@ip-10-0-12-59 backend]$
```

Desde la instancia privada se demuestra que el NGINX se encuentra activo

```
[ec2-user@ip-10-0-12-59 backend]$ curl localhost:3000/health
{"status":"ok"}[ec2-user@ip-10-0-12-59 backend]$
[ec2-user@ip-10-0-12-59 backend]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-11-14 18:44:02 UTC; 1h 53min ago
     Process: 26033 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 26034 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 26035 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Main PID: 26036 (nginx)
      Tasks: 3 (limit: 1053)
     Memory: 3.2M
        CPU: 58ms
    CGroup: /system.slice/nginx.service
            └─26036 "nginx: master process /usr/sbin/nginx"
              └─26037 "nginx: worker process"
                └─26038 "nginx: worker process"
```

El curl hacia localhost muestra el html de login, el cual es el primero al iniciar la app

```
[ec2-user@ip-10-0-12-59 backend]$ curl http://bookstor-ALB-M9JX7wTfrV6p-2072659842.us-east-1.elb.amazonaws.com
curl: (3) URL rejected: No host part in the URL
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Login - Librería</title>
  <link rel="stylesheet" href="css/styles.css">
  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
  <!-- Aquí se define API URL = "/api" -->
  <script src="js/config.js"></script>
</head>
<body>
  <header>
    <h1>📖 Librería Online</h1>
  </header>

  <main>
    <section>
      <h2>Iniciar sesión</h2>
      <form id="form-login">
        <label>Usuario:</label><br>
```

7.7 Pruebas de monitoreo y notificaciones (CloudWatch y SNS)

En el ámbito de monitoreo, se verificó en la consola de Amazon CloudWatch la creación de las alarmas asociadas tanto al ALB como al Auto Scaling Group, por ejemplo, métricas relacionadas con alta utilización de CPU o incremento en los errores 5xx. Para validar su funcionamiento, se generaron condiciones de carga o de fallo en la aplicación, como detener temporalmente el backend, y se observó cómo las métricas cambiaban y las alarmas pasaban

del estado “OK” al estado “ALARM”. Además, en caso de haberse configurado una suscripción de correo electrónico mediante Amazon SNS, se comprobó la recepción de notificaciones asociadas a dichas alarmas. Cuando una métrica superaba el umbral establecido y la alarma cambiaba a estado crítico, se recibía un mensaje en la dirección de correo registrada, confirmando así la correcta integración entre CloudWatch y SNS para las alertas operativas.

```
~ $ aws cloudwatch describe-alarms --region us-east-1

{
  "MetricAlarms": [
    {
      "AlarmName": "bookstore-alb-5xx",
      "AlarmArn": "arn:aws:cloudwatch:us-east-1:344804352841:alarm:bookstore-alb-5xx",
      "AlarmDescription": "ALB 5xx > 5 per 2 minutes",
      "AlarmConfigurationUpdatedTimestamp": "2025-11-14T18:38:46.136000+00:00",
      "ActionsEnabled": true,
      "OKActions": [],
      "AlarmActions": [
        "arn:aws:sns:us-east-1:344804352841:bookstore-alerts"
      ],
      "InsufficientDataActions": [],
      "StateValue": "OK",
      "StateReason": "Threshold Crossed: no datapoints were received for 2 periods and 2 missing datapoints were treated as [NonBreaching].",
      "StateReasonData": "{\n  \"version\": \"1.0\", \"queryDate\": \"2025-11-14T18:40:03.476+0000\", \"statistic\": \"Sum\", \"period\": 60, \"recentDatapoints\": [], \"threshold\": 5.0, \"thresholdType\": \"Upper\", \"metricName\": \"HTTPCode_ELB_5XX_Count\", \"namespace\": \"AWS/ApplicationELB\", \"dimensions\": {\n    {\n      \"Name\": \"LoadBalancer\",",
      "StateUpdatedTimestamp": "2025-11-14T18:40:03.478000+00:00",
      "MetricName": "HTTPCode_ELB_5XX_Count",
      "Namespace": "AWS/ApplicationELB",
      "Statistic": "Sum",
      "Dimensions": [
        {
          "Name": "LoadBalancer",
```

Alarms (3)

☐ Hide Auto Scaling alarms

Clear selection

Create composite alarm

Actions

Create alarm

Alarm state: Any

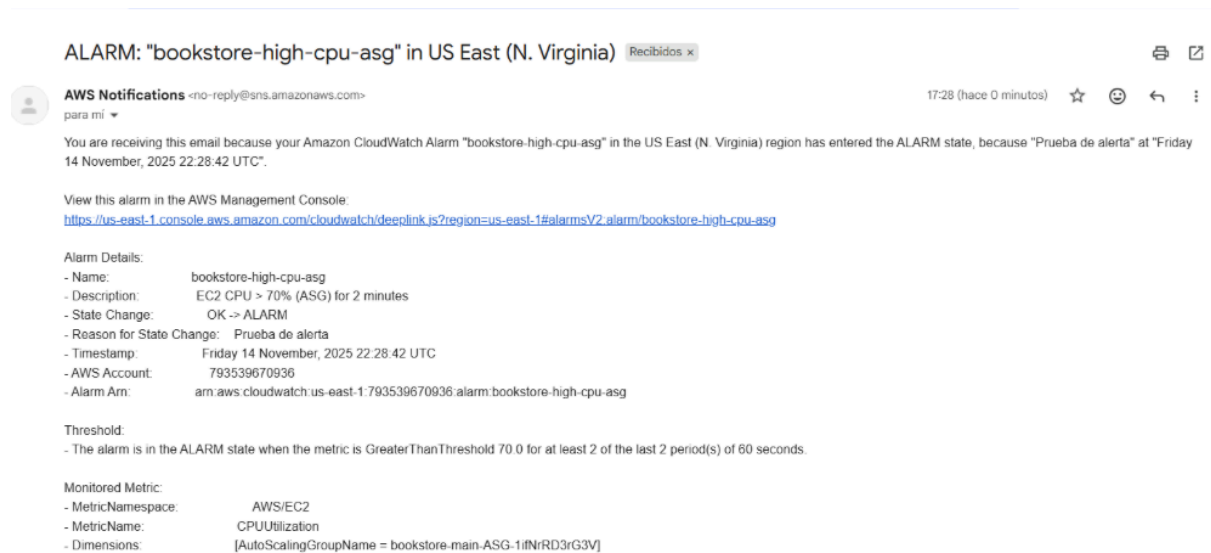
Alarm type: Any

Actions status: Any

< 1 >

<input type="checkbox"/>	Name	State	Last state update (UTC)	Conditions	Actions
<input type="checkbox"/>	bookstore-tg-unhealthy	OK	2025-11-14 18:49:25	UnHealthyHostCount > 0 for 2 datapoints within 2 minutes	Actions enabled Warnin
<input type="checkbox"/>	bookstore-high-cpu-asg	OK	2025-11-14 18:43:44	CPUUtilization > 70 for 2 datapoints within 2 minutes	Actions enabled Warnin
<input type="checkbox"/>	bookstore-alb-5xx	OK	2025-11-14 18:40:03	HTTPCode_ELB_5XX_Count > 5 for 2 datapoints within 2 minutes	Actions enabled Warnin

```
},
{
  "AlarmName": "bookstore-tg-unhealthy",
  "AlarmArn": "arn:aws:cloudwatch:us-east-1:344804352841:alarm:bookstore-tg-unhealthy",
  "AlarmDescription": "Unhealthy targets > 0 for 2 minutes",
  "AlarmConfigurationUpdatedTimestamp": "2025-11-14T18:38:46.442000+00:00",
  "ActionsEnabled": true,
  "OKActions": [],
  "AlarmActions": [
    "arn:aws:sns:us-east-1:344804352841:bookstore-alerts"
  ],
  "InsufficientDataActions": [],
  "StateValue": "OK",
  "StateReason": "Threshold Crossed: 1 datapoint [0.0 (14/11/25 18:46:00)] was not greater than the threshold (0.0).",
  "StateReasonData": "{\n  \"version\": \"1.0\", \"queryDate\": \"2025-11-14T18:49:25.352+0000\", \"start\": \"2025-11-14T18:45:00.000+0000\", \"statistic\": \"Maximum\", \"period\": 60, \"recentDatapoints\": [1.0, 0.0], \"threshold\": 0.0, \"thresholdType\": \"Upper\", \"metricName\": \"UnHealthyHostCount\", \"namespace\": \"AWS/ApplicationELB\", \"dimensions\": {\n    {\n      \"Name\": \"LoadBalancer\",",
  "StateUpdatedTimestamp": "2025-11-14T18:49:25.354000+00:00",
  "MetricName": "UnHealthyHostCount",
  "Namespace": "AWS/ApplicationELB",
  "Statistic": "Maximum",
  "Dimensions": [
    {
      "Name": "LoadBalancer",
```



7.8 Pruebas de auditoría y trazabilidad (CloudTrail y S3)

Se revisó en la consola de AWS CloudTrail la existencia de un trail configurado para registrar las llamadas a la API de la cuenta y se confirmó que dicho trail estaba apuntando al bucket S3 definido en la plantilla de infraestructura. Después se realizaron distintas operaciones sobre los recursos de AWS (creación, modificación y eliminación de componentes de la infraestructura) y pasado un tiempo se revisaron tanto los eventos en la consola de CloudTrail como los archivos generados en el bucket S3. Esta revisión permitió verificar que las acciones administrativas quedaban registradas y almacenadas de forma centralizada, garantizando la trazabilidad de cambios sobre la infraestructura desplegada.

```
~ $ aws cloudtrail describe-trails --region us-east-1

{
  "trailList": [
    {
      "Name": "bookstore-trail",
      "S3BucketName": "bookstore-main-trailbucket-jitpivcqgz5o",
      "S3KeyPrefix": "cloudtrail",
      "IncludeGlobalServiceEvents": true,
      "IsMultiRegionTrail": true,
      "HomeRegion": "us-east-1",
      "TrailARN": "arn:aws:cloudtrail:us-east-1:344804352841:trail/bookstore-trail",
      "LogFileValidationEnabled": true,
      "HasCustomEventSelectors": false,
      "HasInsightSelectors": false,
      "IsOrganizationTrail": false
    }
  ]
}
```