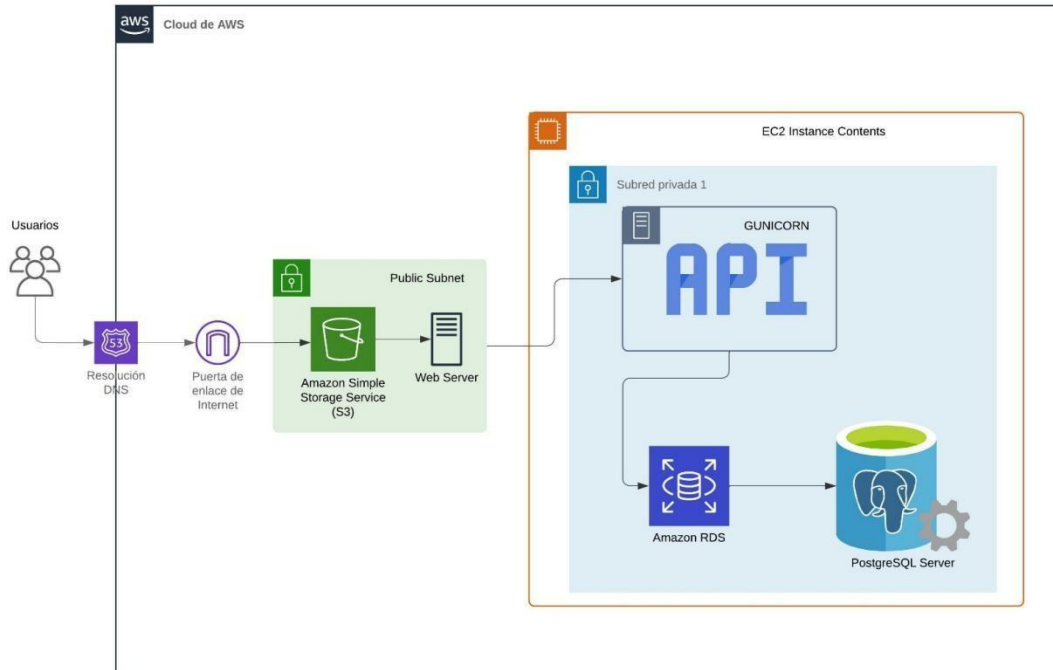


Despliegue en Servidor Virtual con Base de Datos

17-11-2024

Diagrama de despliegue de la arquitectura en AWS



Lo primero que hacemos es crear la instancia de EC2; hemos utilizado Amazon Linux 2023 AMI con la configuración por defecto, pares de clave vockey, t2.micro... El nombre de nuestra instancia es **ec2-practica**.

Adicionalmente, hemos decidido usar el servicio **Elastic IP** para tener una IP fija para nuestra EC2, de modo que no cambie cada vez que se cierra la sesión. La IP es **54.84.202.12**.

Ahora creamos la Base de Datos, para lo cual usamos **Amazon RDS** y elegimos **PostgreSQL**. El nombre de la DB es **rds-practica**, el usuario maestro es **postgres** y su contraseña es **postgres-user**, y el nombre inicial de la DB es **sample**. En cuanto a la **configuración**, hemos usado la versión 15.7, con db.t3.micro, la imagen free tier y conectada a una instancia de EC2, donde seleccionaremos la instancia que acabamos de crear. El resto es la configuración por defecto.

Ahora desde la terminal de la instancia de ec2 podemos comprobar el acceso a la Base de Datos, para ello exportamos la variable local \$DATABASE_URI:

```
export DATABASE_URI=postgresql://postgres:postgres-user@rdspractica.cfmn4tysf6zh.us-east-1.rds.amazonaws.com/sample
```

Ahora ejecutaremos el comando:

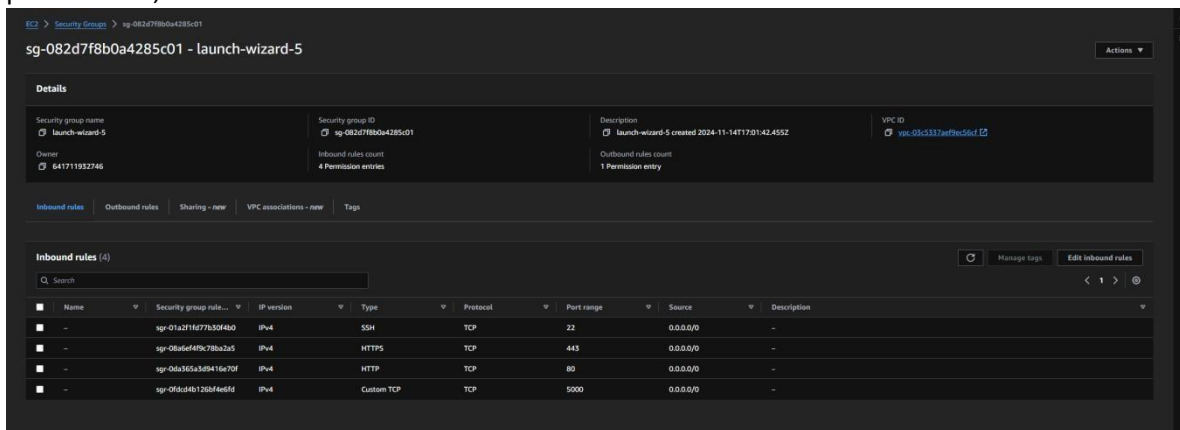
```
psql $DATABASE_URI
```

Si lo hemos hecho bien, nos saldría el **prompt** de la DB que en este caso sería **sample>**.

Ahora tenemos que instalar **pip** e ir a la carpeta del game-store e instalar los requerimientos (para ello antes hay que pasar la carpeta al EC2 con scp), y en caso de que no esté instalado **gunicorn**, también habría que hacerlo para poder lanzar el demonio de la API REST:

```
sudo dnf install python3-pip
pip install -r requeriments.txt
pip install gunicorn
```

Ahora tenemos que modificar el grupo de seguridad de la INSTANCIA para habilitar la escucha en el puerto 5000, en nuestro caso es el **launch-wizard-5**:



Ahora ejecutamos el daemon con el siguiente mandato:

```
gunicorn --bind 0.0.0.0:5000 initAlchemy:api --daemon
```

Con el comando `sudo netstat -tln | grep 5000` podemos comprobar que el puerto 5000 está en escucha y con `ps aux | grep gunicorn` podemos comprobar que el daemon funciona.

Ahora creamos un bucket con S3 para el swagger con la configuracion por defecto: **s3-practica**

Una vez creado le pasamos los archivos del swagger, pero antes hay que cambiar la URL en el **swaggerconfig.yaml**.

Pondremos `http://ec2-54-84-202-12.compute-1.amazonaws.com:5000/api`

```
! swagger-config.yaml X
C:\Users> angel > Downloads > material > material-pra-1 > swagger > ! swagger-config.yaml > servers
1  openapi: 3.0.3
2  info:
3    title: 'Swagger: Game Store'
4    description: |-
5      This is a sample Game Store Server based on the OpenAPI 3.0 specification. You can find out more about
6      Swagger at https://swagger.io(https://swagger.io).
7
8    contact:
9      email: a.mcarro@upm.es
10   license:
11     name: Apache 2.0
12     url: http://www.apache.org/licenses/LICENSE-2.0.html
13   version: 1.0.0
14 servers:
15   - url: http://localhost:8000/api
16   - url: http://54.84.202.12:5000/api
17 tags:
18   - name: game
19     description: Everything about Games
20   - name: store
21     description: Access to store orders
22   - name: user
23     description: Operations about user
24   - name: tag
25     description: All about Tags
26   - name: category
27     description: All about Categories
```

En este caso, la API está en el puerto 5000 de nuestra instancia EC2. Para pasar los archivos simplemente clickamos en el bucket y le damos a upload.

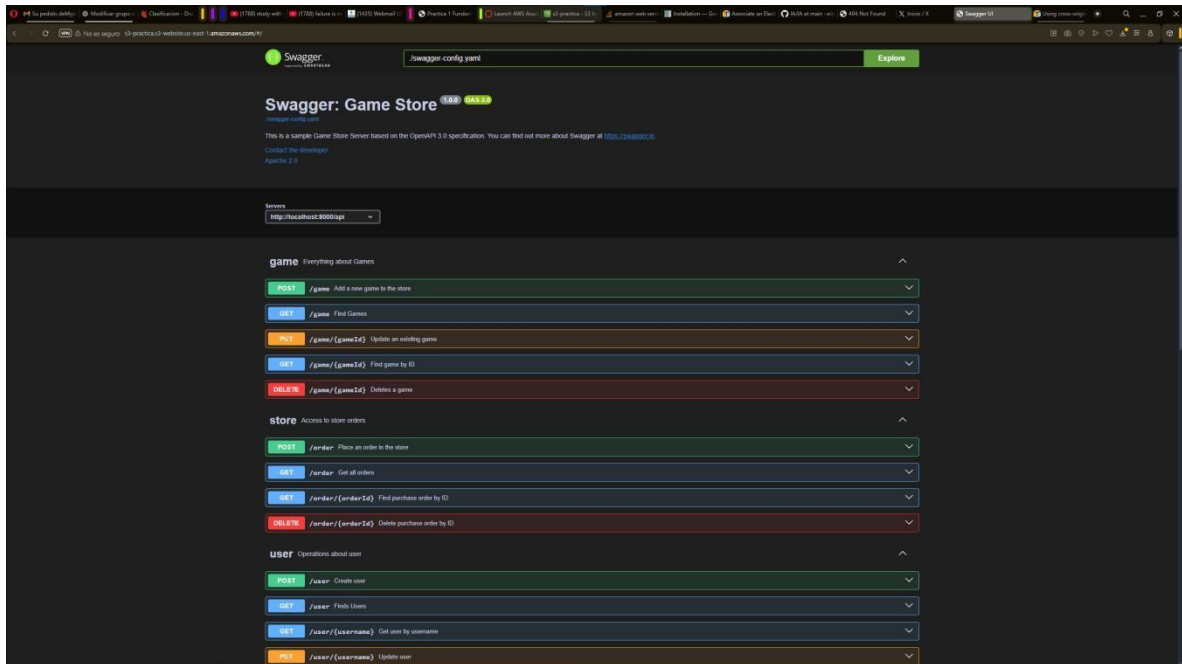
Si no lo hemos hecho antes, tenemos que permitir que el bucket aloje una web. Para ello, vamos a **static website hosting**, le damos a enable y ponemos el fichero **index.html** como index document.

Una vez hecho esto, tenemos que configurar el acceso a la web. Para ello, desactivamos el bloqueo de todo el acceso público, y en la política del bucket ponemos:

```
{
  "Version": "2012-10-17",
  "Statement": [ {
    "Sid": "PublicReadGetObject",
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::s3-practica/*"
  }
]
```

Esto permite el acceso a cualquier usuario leer el contenido del bucket. Ahora ponemos la siguiente URL en el navegador y estamos dentro:

`http://s3-practica.s3-website.us-east-1.amazonaws.com`



Ahora desde aquí podemos hacer las pruebas, donde pone **servers** eligiremos el que tiene el mismo URL que pusimos anteriormente en el **swagger-config.yaml**.

Casos de prueba:

1. Buscar el juego con id=2:

GET

/game/{gameId} Find game by ID

⬆

Returns a single game

Parameters

Cancel

Name	Description
gameId * required integer(\$int64) (path)	ID of game to return

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://ec2-54-84-202-12.compute-1.amazonaws.com:5000/api/game/2' \
-H 'accept: application/json'
```

Request URL

```
http://ec2-54-84-202-12.compute-1.amazonaws.com:5000/api/game/2
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "availableQuantity": 10, "category": { "id": 2, "name": "PlayStation 5" }, "id": 2, "name": "Final Fantasy VII Remake", "imageUrl": "https://m.media-amazon.com/images/I/81MBCAno24L._AC_SX300_SY300_QL70_ML2_.jpg", "tags": [{ "id": 2, "name": "RPG" }] }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 232 content-type: application/json</pre></div></div>

Responses

Code	Description	Links
200	successful operation	No links

Media type

application/json

Controls Accept header.

Example Value

Schema

```
{
  "id": 1,
  "name": "The Legend of Zelda: Tears of the Kingdom",
  "availableQuantity": 10,
  "category": {
    "name": "Nintendo Switch"
  },
  "imageUrl": "https://m.media-amazon.com/images/I/711QbcZ5LFL._AC_SX522_.jpg",
  "tags": [
    {
      "name": "RPG"
    }
  ]
}
```

400

Invalid ID supplied

No links

404

Game not found

No links

2. Buscamos el juego Final Fantasy VII Remake:

GET

/game

Find Games

⌵

Returns all games in the store, filtered by the patter given in parameters

Parameters

Cancel

Name	Description
pattern	Value that need to be considered for filtering the names of the games
string (query)	<input type="text" value="Final Fantasy VII Remake"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://ec2-54-84-202-12.compute-1.amazonaws.com:5000/api/game?pattern=FinalX20FantasyX20VIIx20Remake' \
  -H 'accept: application/json'
```

Request URL

```
http://ec2-54-84-202-12.compute-1.amazonaws.com:5000/api/game?pattern=FinalX20FantasyX20VIIx20Remake
```

Server response

Code	Details
200	<div>Response body</div> <div><pre>{ "availableQuantity": 10, "category": { "id": 1, "name": "PlayStation 5" }, "id": 7, "name": "Final Fantasy VII Remake", "photoUrl": "https://m.media-amazon.com/images/I/81M8CAno24L._AC_SX300_SY300_QL70_ML2_.jpg", "tags": [{ "id": 1, "name": "RPG" }] }</pre></div> <div>Response headers</div> <div><pre>content-length: 234 content-type: application/json</pre></div>

Responses

Code	Description	Links
200	successful operation	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": 1,
  "name": "The Legend of Zelda: Tears of the Kingdom",
  "availableQuantity": 10,
  "category": {
    "name": "Nintendo Switch"
  },
  "photoUrl": "https://m.media-amazon.com/images/I/711Qbe25LFL._AC_SX522_.jpg",
  "tags": [
    {
      "name": "RPG"
    }
  ]
}
```

3. Añadimos el juego Grand Theft Auto VI y lo buscamos:

POST

/game Add a new game to the store

⬆

Add a new game to the store

Parameters

Cancel

Reset

No parameters

Request body

application/json

Create a new game in the store

```
{
  "name": "Grand Theft Auto V",
  "availableQuantity": 15,
  "category": {
    "id": 1,
    "name": "PlayStation 5"
  },
  "photoUrl": "https://cdn.hobbyconsoles.com/sites/revi.sxalpringer.es/public/media/image/2020/05/gta-vi-1947857.jpg?tf=1280x1280",
  "tags": [
    {
      "name": "RPG"
    }
  ]
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  http://ec2-54-84-202-13.compute-1.amazonaws.com:5000/api/game' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "Grand Theft Auto V",
    "availableQuantity": 15,
    "category": {
      "name": "PlayStation 5"
    },
    "photoUrl": "https://cdn.hobbyconsoles.com/sites/revi.sxalpringer.es/public/media/image/2020/05/gta-vi-1947857.jpg?tf=1280x1280",
    "tags": [
      {
        "name": "RPG"
      }
    ]
  }'
```

Request URL

http://ec2-54-84-202-13.compute-1.amazonaws.com:5000/api/game

Server response

Code

Details

200

Response body

```
{
  "availableQuantity": 15,
  "category": {
    "id": 1,
    "name": "PlayStation 5"
  },
  "id": 1,
  "name": "Grand Theft Auto V",
  "photoUrl": "https://cdn.hobbyconsoles.com/sites/revi.sxalpringer.es/public/media/image/2020/05/gta-vi-1947857.jpg?tf=1280x1280",
  "tags": [
    {
      "id": 1,
      "name": "RPG"
    }
  ]
}
```

Download

Response headers

```
content-length: 366
content-type: application/json
```

Responses

Code

Description

Links

200

Successful operation

No links

Media type

application/json

Content Accept header

Example Value | Schema

```
{
  "id": 1,
  "name": "The Legend of Zelda: Tears of the Kingdom",
  "availableQuantity": 30,
  "category": {
    "name": "Nintendo Switch"
  },
  "photoUrl": "https://a.media-amazon.com/images/I/71lQw25LFL._AC_SX522_.jpg",
  "tags": [
    {
      "name": "RPG"
    }
  ]
}
```

400

Bad request

No links

415

Unsupported media type

No links

GET

/game Find Games

⬆

Returns all games in the store, filtered by the patter given in parameters

Parameters

Cancel

Name	Description
pattern	Value that need to be considered for filtering the names of the games
string (query)	<div>Grand Theft Auto VI</div>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://ec2-54-84-202-12.compute-1.amazonaws.com:5000/api/game?pattern=GrandTheftAutoVI' \
  -H 'accept: application/json'
```

Request URL

```
http://ec2-54-84-202-12.compute-1.amazonaws.com:5000/api/game?pattern=GrandTheftAutoVI
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "availableQuantity": 10, "category": { "id": 1, "name": "PlayStation 5" }, "id": 7, "name": "Grand Theft Auto VI", "photoUrl": "https://cdn.hobbyconsolas.com/sites/navi.axelspringer.es/public/media/image/2020/05/gta-vi-1947857.jpg?tf=1200x1200", "tags": [{ "id": 1, "name": "RPG" }] }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 266 content-type: application/json</pre></div></div>

Responses

Code	Description	Links
200	<div>successful operation</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "id": 1, "name": "The Legend of Zelda: Tears of the Kingdom", "availableQuantity": 20, "category": { "name": "Nintendo Switch" }, "photoUrl": "https://a.media-amazon.com/images/I/71lQbeZSLFl._AC_SX522_.jpg", "tags": [{ "name": "RPG" }] }</pre></div>	

 No links |